



# ASG-JOB/SCAN Documentation

Version: 8.0.1

Publication Date: October, 2020

©2020 ASG Technologies Group, Inc. All rights reserved.

---

The information contained herein is the confidential and proprietary information of ASG Technologies Group, Inc. Unauthorized use of this information and disclosure to third parties is expressly prohibited. This technical publication may not be reproduced in whole or in part, by any means, without the express written consent of ASG Technologies Group, Inc.

All names and products contained herein are the trademarks or registered trademarks of their respective holders.

ASG Technologies Worldwide Headquarters Naples Florida USA | [asg.com](http://asg.com) | [info@asg.com](mailto:info@asg.com)

708 Goodlette Road North, Naples, Florida 34102 USA Tel: 239.435.2200 Fax: 239.263.3692 Toll Free: 800.932.5536 (USA only)

# Contents

---

<b>Preface</b> .....	<b>xvii</b>
About this Publication .....	xvii
Publication Conventions .....	xvii
Worldwide Customer Support .....	xvii
ASG Customer Care Expert Solution Services Portal (ACCESS) .....	xviii
Product Support Policy .....	xviii
ASG Documentation/Product Enhancements .....	xix
<b>What's New?</b> .....	<b>1</b>
<b>Release Notes</b> .....	<b>3</b>
<b>Glossary</b> .....	<b>7</b>
<b>Installation</b> .....	<b>16</b>
Installing the Product .....	17
Planning .....	17
Product Installation Options .....	18
Downloading and Transferring the Product Files .....	18
Installing the Product .....	22
Installing the TSO Online Facility .....	30
Adding JOB/SCAN to the ISPF Primary Menu .....	34
Allocating the Files .....	36
Applying Product Authorization Code(s) .....	36
Creating the JOB/SCAN Environment .....	39
Step 1—Modify Detail Statements .....	39
Step 2—Specify PROCLIBs in JSOPT02 .....	40
Step 3—Modify Run and Report Options in JSOPT02 .....	41
Step 4—Modify Control Tables in JSOPT02 .....	42
Step 5—Modify the Batch PROC .....	45
Step 6—Install Edit Macros .....	45
Step 7—Set Up the Product Usage Facility .....	46
Setting Up Optional Interfaces and Exits .....	50
General Considerations for User Exits .....	50
User Exit Message Format .....	51
JCL Standards Facility .....	52

---

Selection Exit Interface .....	53
Security Exit Interface .....	55
Library Management Interfaces .....	60
Database Interfaces .....	60
ROSCOE Interface .....	63
Other Interfaces .....	66
Final Installation Tasks .....	68
Special Considerations .....	68
Initializing JOB/SCAN .....	68
Running the Product in an APF-authorized Environment .....	76
When is an APF-authorized Environment Required? .....	76
Setting Up the ISPF Environment .....	76
Setting Up Batch Processing .....	77
Security Exit Considerations .....	77
Enabling DOCU/TEXT .....	78
Installing the TSO Online Facility .....	78
Adding DOCU/TEXT to the ISPF Primary Menu .....	80
Allocating the Files .....	82
Applying Product Authorization Code(s) .....	82
Running DOCU/TEXT in an APF-authorized Environment .....	82
Creating the DOCU/TEXT Environment .....	83
Allocating PDS Storage .....	89
Setting Up Optional Interfaces and Exits .....	93
Final Installation Tasks .....	105
Calling JOB/SCAN from a User Program .....	107
<b>System Administrator Tasks .....</b>	<b>108</b>
Administration Basics .....	109
System Administrator Role .....	109
Using the Product Administration Panel .....	109
Options Overview .....	111
Administering JOB/SCAN .....	114
Assigning the Product Administrator Role .....	114
Controlling Access to JOB/SCAN .....	115
Managing JOB/SCAN through Options Members .....	117

---

Supplementing SJJSPARM Options .....	124
Allowing Users to Create Lists of Jobs .....	127
Allowing Users to Create Standards Programs .....	127
Adding Items to the Standards Utility Menu Panel .....	128
Increasing File Space Allocation .....	128
Adding Site-Defined Functions .....	129
Managing Database Interfaces .....	130
Managing Scheduler Interfaces .....	133
Managing Security Interfaces .....	137
Optimizing JOB/SCAN's Resource Requirements .....	141
Ensuring Equivalent JCL Validation Results .....	145
Managing JOB/SCAN's Output Messages .....	152
Options .....	158
System Identification Group .....	158
Access Control Group .....	159
File Allocations Group .....	168
Run Options Group .....	175
Report Option Group .....	177
Control Table Group .....	184
Input Selection Group .....	187
JOB/SCAN Libraries .....	192
SJJSPARM Library .....	192
SJJSSAMP Library .....	193
SJJSSTD Library .....	196
SJJSDemo Library .....	198
Product Usage Facility .....	200
Creating and Maintaining a Product Usage File .....	201
Activating and the Product Usage Facility .....	201
Managing Collection Periods .....	203
Displaying Site-wide Product Usage Information .....	206
Displaying Product Usage by Period .....	207
Displaying Usage Data by Individual .....	210
Managing Users .....	212
Exporting Usage Data .....	214
<b>User Information - JOB/SCAN .....</b>	<b>223</b>

---

---

Introduction to JOB/SCAN .....	224
JOB/SCAN Basics .....	224
Initializing JOB/SCAN .....	228
JCL Validation .....	237
The Basic Validation Process .....	237
Validating JCL from the Panels .....	237
Validating JCL with Edit Macros .....	243
Validating JCL in Batch .....	249
Advanced Topics for JCL Validation .....	253
Selecting Members .....	253
SJJSPARM Library Options Members .....	267
Effects of Options on Validation Results .....	272
Specifying a First Proclib for JCL Validation .....	274
Searching a PROC .....	276
Validating JCL that Includes CA-7 #J Statements .....	277
Reports .....	279
Structured JCL Listing (SJL) .....	279
Detail Information Section of the SJL .....	281
Error Summary Section of the SJL .....	285
Viewing Options in Effect .....	286
Suppressing JOB/SCAN Error Messages .....	288
Displaying Extended Error Messages .....	291
JMSG: Extended Error Messages for the JEM Edit Macro .....	293
JCL Change .....	296
Basic Concepts .....	296
Creating Change Command Members .....	297
Overall Command Syntax .....	297
JCL Change Commands Order of Application .....	299
Selector Clause (ON Command) .....	301
Verb Clause .....	302
Setting the Scope .....	309
Basic Change Examples .....	314
Running JCL Change .....	325
Verifying Change Commands .....	330

---

Advanced Features .....	331
JCL Reformatter .....	341
Overview .....	341
Setting Up Reformatter through Panels .....	343
Creating Reformatter Command Members .....	347
Writing Reformatter Commands .....	348
Accessing Reformatter through Tables .....	351
Running Reformatter in Foreground .....	354
Running Reformatter with JREF Edit Macro .....	358
Reformatting JCL with JCHG .....	359
Running Reformatter in Batch Mode .....	360
Reformatting Comments with the BC Table .....	362
Removing Sequence Numbers from Columns 73-80 .....	366
Commenting a Reformat Member .....	367
ROSCOE Interface .....	368
ROSCOE Interface Facilities .....	368
ROSCOE RPF Facility and ISPF Functional Differences .....	369
Execution of JOB/SCAN RPF Panels .....	370
ROSCOE RPF Panels .....	370
ETSO Overview .....	374
ETSO RPFs .....	376
<b>User Information - JOB/SCAN with Schedulers .....</b>	<b>377</b>
User Information - JOB/SCAN with CA-7 .....	378
Getting Started .....	378
Using JCA to Validate JCL .....	379
Validating Jobs from Foreground .....	381
Using CA7SCHD to Validate a CA-7 Schedule .....	383
Using JSCA7 to Validate a JOB .....	385
JOB/SCAN CA-7 Interface Setup .....	387
User Information - JOB/SCAN with ESP .....	390
Getting Started .....	390
JOB/SCAN-ESP Basic Process .....	390
Usage Scenarios .....	392
Terminology .....	392
Validating Individual Members .....	393

---

---

Validate a Single Member with the JSESP Edit Macro .....	394
Validate a Single Member in Batch .....	398
Validate Members from the Validation Panel .....	399
Validating an ESP Event .....	401
Validate an Event with the TSO JSEVENT Command .....	402
Validate an ESP Event in Batch .....	407
Validate an Event with the ESP JCLSCAN Exit .....	409
Validating a Schedule .....	414
Installing ESP .....	415
Modify the CLIST Library (SJJSCLIST) Members .....	416
Activate the ESP Scheduler Interface .....	416
Verify Installation .....	418
User Information - JOB/SCAN with TWS .....	423
Getting Started .....	423
JOB/SCAN TWS Interface Overview .....	423
JOB/SCAN TWS Interface Basic Usage Concepts .....	424
JOB/SCAN Commands for TWS .....	426
JVAL Command .....	426
From the ASG-JOB/SCAN Primary Menu .....	427
Validate JCL for a Single Member .....	427
JTWS Edit Macro Syntax and Parameters .....	428
Using the JTWS EDIT Macro in ISPF Edit or View .....	428
Using the JTWS EDIT Macro from TWS Panels .....	431
Validate Members from the Validation Panel .....	438
Validate a Single Member in Batch .....	440
Validating a Plan .....	442
Validate JCL for the Current Plan .....	442
Validate JCL for the Trial Plan .....	445
Validate Application JCL .....	447
Validate Application JCL from TWS .....	447
Application JCL Validation from JOB/SCAN .....	450
Installation .....	452
Installation Checklist .....	452
Security .....	452

---

Installation Procedure .....	453
Providing Access to JOB/SCAN for TWS .....	454
Activate TWS Scheduler Interface .....	462
Reporting on TWS Special Resources (Optional) .....	464
Verify Installation .....	465
JCL Parameters for the JVAL Command JCL .....	468
Preserving Access to an Existing JCL Edit Tool .....	470
<b>Reference Information .....</b>	<b>472</b>
Run Option Statements .....	473
Access Control Cards (ACC) .....	474
Catalog Access (CAC) .....	475
CONTROL-M Exits (CMX) .....	476
CONTROL-M Variables (CMV) .....	477
DBCS .....	478
Default Unit Names (DUN) .....	479
Device Codes (DVC) .....	479
DMS Volume Serial (DMSV) .....	480
Exec Parm String (EPS) .....	481
GDG Processing (GDGP) .....	482
Group ID (GRP) .....	482
Headings (HDG) .....	483
HSM Volume Serial (HSMV) .....	483
ISPF Specifications (ISS) .....	484
JCL Change Parameters (JCLCP) .....	485
JES2 Command Character (J2C) .....	486
JLJPA PROCLIB Type (JPAPT) .....	486
JLJPC PROCLIB Type (JPCPT) .....	487
Job Limit .....	488
JOB/SCAN PROC (JSP) .....	489
LIBRARIAN (LIBR) .....	490
LINECNT .....	491
Migrate Device Type (MDTP) .....	491
Operating System Level (OSL) .....	492
Output Specifications (OSS) .....	494
Override Indicator (ORI) .....	495

---



---

Prefix (PFX) .....	495
Processing Specifications (PSS) .....	496
Program Product Interface (PPI) .....	497
Report Heading .....	500
Resolve System Symbols (RSS) .....	500
Scan JCL Only (SJO) .....	501
Scheduler (SCHED) .....	502
Security Exit .....	503
SEMRESET .....	504
Separator (SEP) .....	505
Short Descriptions (SDS) .....	505
SMS .....	509
Span Members (SMB) .....	510
Standards Name (STN) .....	511
Subsystem (SSS) .....	512
Suppress Error Messages (SEM) .....	513
Updated JCL File Type (UJFT) .....	514
Use Error Table (UET) .....	514
Use PROC Exclude Table (UPCX) .....	515
User ID (UID) .....	515
Write Selected JCL (WSJ) .....	516
Control Tables .....	518
Control Table-interaction with Run Options .....	519
Control Table AA: Device Type, Device Group, and Unit Address .....	519
Control Table AB: Control Card DD Name .....	521
Control Table AC: Input/Output DD Names .....	522
Control Table AD: Utility Programs .....	523
Control Table AM: Tape Management Locations (Out-of-Area codes) .....	524
Control Table AN: Procedures Excluded from Validation .....	526
Control Table AP: PARM Programs .....	527
Control Table AQ: Control Card Dataset Names .....	528
Control Table AR: Return Code Simulation .....	528
Control Table AS: Nonstandard JES Statements .....	529
Control Table AT: Nonstandard OUTPUT Keywords .....	530

Control Table AU: Job Steps to Ignore .....	531
Control Table AW: Suppress JCL Error Messages .....	532
Control Table AX: Nonstandard Parameters for DD Statements .....	533
Control Table AY: Nonstandard Parameters for EXEC Statements .....	534
Control Table AZ: Nonstandard Parameters for Job Statements .....	534
Control Table BA: Reformatting DD Statements .....	535
Control Table BB: Reformatting EXEC Statements .....	537
Control Table BC: Additional Reformatter Options .....	538
Control Table BD: Reformatting PROC Statements .....	543
Control Table BE: Reformatting JOB Statements .....	544
Control Table BF: Reformatting OUTPUT Statements .....	546
Control Table BG: Reformatting SET Statements .....	548
Control Table BH: Reformatting JCLLIB Statements .....	549
Control Table BI: Reformatting INCLUDE Statements .....	550
Control Table CB: Program Name Exit .....	550
Control Table CC: VOLSER .....	551
Control Table CD: DB2 Group Attachment Names .....	552
Control Table CE: Scheduler Restart Program Names .....	552
Control Table CF: Subsystem Dataset Disposition .....	553
Control Table CG: Datasets That Are Not Control Card Datasets .....	553
Control Table HF: HFS Path Name DD statements .....	554
Control Table Ux: Standards Facility User Table .....	554
Scheduler Interfaces .....	556
CA-7 Scheduler Interface .....	556
Handling #J Cards with the CA-7 Scheduler Interface .....	557
Validating Jobs in Run-time Order with the CA-7 Scheduler Interface .....	558
CA-Scheduler Interface .....	559
CA-Driver Interface .....	561
CONTROL-M Scheduler Interface .....	562
DCMS Scheduler Interface .....	567
ESP Scheduler Interface .....	568
JOBTRAC Scheduler Interface .....	569
TWS z/OS Scheduler Interface .....	572
ASG-Zeke Scheduler Interface: \$ Symbolic Parameters .....	572
BETA42 Scheduler Interface .....	575

Security Interfaces .....	577
Security Interface Overview .....	577
Security Interface Functions .....	578
Security Interface Description .....	581
Other Interfaces .....	585
CA-PANVALET Interface .....	585
Running Jobs in an APF-Authorized Environment .....	585
CA-1 (UCC1) Tape Management System .....	587
ChangeMan Interface .....	588
DB2 Interface .....	590
ENDEVOR Interface .....	591
IMS (Information Management System) Interface .....	591
CA-DYNAM/TLMS Interface .....	593
ZARA Interface .....	594
DFSMSrmm – Removable Media Manager .....	594
CA-11 Interface .....	595
CONTROL-R Interface .....	595
SCLM Interface .....	596
Extended Parameter List Samples .....	598
Program Name Exit .....	603
Setting Up the Program Name Exit .....	603
Program Name Exit: How It Works .....	603
User Abends Issued by JOB/SCAN .....	607
Selection Exit Interface .....	609
Overview .....	609
Selection Exit Defined In Edit Macros .....	613
INSJOB: Selection Exit to Insert a Jobcard .....	613
JOBUSER: Selection Exit to Insert 'USER=' Parameter .....	614
<b>Standards Information .....</b>	<b>615</b>
Standards Overview .....	616
Introduction to Standards .....	616
The Standards Process .....	618
Using the Standards Facility .....	620
Using Standards in ISPF Mode .....	620

---

Using Standards with Edit Macros .....	621
Using Standards in Batch Mode .....	621
Standards Facility Error Messages .....	623
Standards Error Messages .....	623
Default Error Message Handling .....	625
Extended Error Messages .....	625
Individual Control Over Standards Messages .....	627
Changing or Suppressing JOB/SCAN Error Messages .....	627
Using Extended Parameter List Fields and Tables .....	628
EPL Example .....	628
EPL Example Fields .....	628
EPL Documentation .....	629
Table Processing .....	629
Using Control Table Entries .....	631
COBOL Standards Implementation Using the Prompter .....	633
Prompter-Generated Standards .....	633
Getting Started with the Prompter .....	635
Creating and Maintaining Standards with the Prompter .....	639
Preparing Standards Program for Execution .....	644
COBOL Primer .....	646
Introduction to COBOL .....	647
Words and Symbols .....	647
Statements .....	648
Sentences .....	648
Comments .....	649
Paragraphs and Sections .....	649
Standards Program Sections .....	649
Program Divisions .....	650
Format of a Standard .....	650
Condition .....	650
Message and Message Number .....	650
Severity Code .....	651
Z999-SET-ERROR .....	651
Indentation .....	651
Simple Comparisons (=, >, <, NOT) .....	651

---

---

Testing Single Character Fields .....	652
Testing CLASS .....	652
The Difference Between 'AS CODED' and 'AS USED' EPL Fields .....	652
Compound Comparisons .....	653
Using AND and OR .....	654
Testing Two Different JCL Fields .....	654
Testing One Character in a Multi-Character Field .....	654
Testing a Group of Characters in a Multi-Character Field .....	655
Testing a Group of Characters in a Field .....	656
Using the SUBSTRING Functions .....	657
Testing Numeric and Character Fields .....	658
Testing JCL Parameters .....	659
Adding Fields to WORKING-STORAGE .....	660
Field Names .....	660
Level Numbers .....	660
Pictures .....	661
Testing Dataset Name Nodes .....	662
Using UNSTRING .....	663
Counting Items .....	663
Standards Requiring Calculations .....	665
Using the CALL-*SECTIONS .....	666
Multiple IFs and ELSEs .....	668
Defining External Tables .....	669
Using Dynamic Tables .....	671
Advanced Topics .....	673
Arithmetic .....	676
Generating JCL Change Commands with the Standards Program .....	678
COBOL Standards Testing and Debugging .....	681
Finding and Correcting Compiler Errors .....	681
Setting Up Test JCL Scenarios .....	682
Values Passed to Standards .....	683
Section or Statement Execution .....	686
REXX Standards .....	688
Overview of REXX Standards Programs .....	688

Additional REXX Functions .....	688
REXX Documentation .....	688
REXX Standards Implementation .....	690
Setting up the REXX Environment with Required DDnames .....	690
Specifying the Name of REXX Standards .....	690
Writing a REXX Standard .....	691
Sample Standards Statements .....	692
Table of JCL Call Types .....	693
Using Data Elements from the EPL .....	695
EPL Field Examples .....	696
Table Processing Overview and Procedure .....	697
REXX Programming Notes .....	698
Capitalization .....	698
Input Parameters .....	698
CALL_TYPE Processing Subroutines .....	698
User Control Table Processing .....	699
Error Message Table Processing .....	700
Control Card Table Processing .....	700
Individual Control Over Messages .....	701
Examining a Dataset Name .....	702
Alternate Method for Examining a Dataset Name .....	703
Global Variables .....	703
Host Command Environment .....	705
Compiling REXX Standards Programs .....	706
Performance Considerations .....	706
REXX Standards Errors and Abends .....	708
DSS08nn Error Messages .....	708
RC(-3) Return Codes .....	709
User Abends .....	710
IRX0040I Error Messages .....	711
REXX Functions .....	712
JSGET .....	712
JSPUT .....	713
JSMSG .....	714
JSTABLE .....	717

---

JSDUMP .....	720
SJJSSTD Library Members .....	721
Call Types .....	723
Common Questions about Writing Standards .....	726
<b>Messages and Codes .....</b>	<b>729</b>
System Messages .....	730
Return Codes .....	892
<b>ASG-JOB/SCAN Plug-in for Eclipse IDE .....</b>	<b>893</b>
Introduction .....	894
Installation .....	895
Installation Requirements .....	895
Installing ASG-JOB/SCAN Plug-in .....	895
Verifying the Installation .....	900
Verifying the z/OSMF Configuration .....	903
Verifying the Liberty Server Configuration .....	905
User Interface .....	907
Connecting to z/OS and ASG-JOB/SCAN .....	917
Setting the JCL Scan Options .....	922
Setting the Processing Options .....	926
Working with z/OS Datasets .....	933
Displaying Datasets .....	933
Editing and Saving Datasets .....	936
Refreshing Datasets .....	938
Working with Local Files .....	939
Displaying Local Files .....	939
Editing and Saving Local Files .....	945
Validating JCL .....	947
Scan JCL .....	947
Scan and submit JCL .....	950
Reports .....	953
Markers .....	958
Show or Hide Markers .....	961
Keyboard Shortcuts .....	965
Customizing Keyboard Shortcuts .....	965

---

ASG-JOB/SCAN REST API Documentation .....	968
Uninstallation .....	969
<b>ASG-JOB/SCAN Plug-in for IBM Explorer for z/OS (z/OS Explorer) .....</b>	<b>973</b>
Introduction .....	974
Installation .....	975
Installation Requirements .....	975
Installing ASG-JOB/SCAN Plug-in .....	975
Verifying the Installation .....	980
Verifying the z/OSMF Configuration .....	983
Verifying the Liberty Server Configuration .....	984
User Interface .....	986
Connecting to z/OS from Remote Systems .....	997
Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator .....	1002
Setting the JCL Scan Options .....	1007
Setting the Processing Options .....	1011
Working with z/OS Datasets .....	1018
Displaying Datasets .....	1018
Editing and Saving Datasets .....	1023
Refreshing Dataset .....	1025
Working with Local Files .....	1027
Displaying Local Files .....	1027
Editing and Saving Local Files .....	1033
Validating JCL .....	1035
Scan JCL .....	1035
Scan and submit JCL .....	1038
Reports .....	1042
Markers .....	1047
Show or Hide Markers .....	1049
Keyboard Shortcuts .....	1053
Customizing Keyboard Shortcuts .....	1053
ASG-JOB/SCAN REST API Documentation .....	1056
Uninstallation .....	1057
<b>ASG-JOB/SCAN Plug-in for Topaz Workbench .....</b>	<b>1060</b>
Introduction .....	1061
Installation .....	1062

---



---

Installation Requirements .....	1062
Installing ASG-JOB/SCAN Plug-in .....	1062
Verifying the Installation .....	1067
Verifying the z/OSMF Configuration .....	1070
Verifying the Liberty Server Configuration .....	1071
User Interface .....	1073
Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator .....	1084
Connecting to z/OS from Host Explorer .....	1089
Setting the JCL Scan Options .....	1094
Setting the Processing Options .....	1098
Working with z/OS Datasets .....	1105
Displaying Datasets .....	1105
Editing and Saving Datasets .....	1112
Refreshing Dataset .....	1114
Working with Local Files .....	1115
Displaying Local Files .....	1115
Editing and Saving Local Files .....	1122
Validating JCL .....	1124
Scan JCL .....	1124
Scan and submit JCL .....	1127
Reports .....	1130
Markers .....	1135
Show or Hide Markers .....	1137
Keyboard Shortcuts .....	1142
Customizing Keyboard Shortcuts .....	1142
ASG-JOB/SCAN REST API Documentation .....	1146
Uninstallation .....	1147

# Preface

This documentation describes how to use our main ASG product JOB/SCAN.

## About this Publication

This documentation describes how to use the different modules that form the ASG product JOB/SCAN.

## Publication Conventions

ASG uses these conventions in technical publications:

Convention	Usage
<b>Bold</b>	Indicates that case-sensitive usage is required for a directory, path, file, dataset, member, database, program, command, or parameter name.
Example:	Verify the settings in the <b>asg.config</b> file.
Capitalization	For system element names, this varies according to the product interface and its operating environment. Mainframe file names use uppercase, for example: Allocate a JSOPTM member in the JLRCL library. Windows file names use mixed cases, for example: Create a text file named SECLIST.txt in the C:\Program Files\ASG\config directory. UNIX file names use mixed case
<i>lowercase italic monospace</i>	Information that you provide according to your particular situation. For example, you would replace filename with the actual name of the file.
Monospace	Characters you must type exactly as they are shown, such as code, JCL, file listings, or command/statement syntax. Also used for denoting brief examples in a paragraph.

## Worldwide Customer Support

ASG provides support throughout the world to resolve questions or problems regarding installation, operation, or use of our products. ASG provides all levels of support during normal business hours and emergency support during non-business hours.

You can access support information from ASG's Support page.

ASG Third-party Support. ASG provides software products that run in a number of third-party vendor environments. Support for all non-ASG products is the responsibility of the respective vendor. In the event a vendor discontinues support for a hardware and/or software product, ASG cannot be held responsible for problems arising from the use of that unsupported version.

## ASG Customer Care Expert Solution Services Portal (ACCESS)

The ACCESS Portal provides online support. Log on to the ACCESS with this information:

Email ID = NNNNNNNNN

Site Password = XXXXXXXXXXX

where:

NNNNNNNNN is your email ID supplied by ASG Product Distribution.

XXXXXXXXXXX is your unique site password supplied by ASG Product Distribution.

If you do not have your logon information, contact your local support center. This table outlines the support response times you can expect:

Severity	Meaning	Expected Support Response Time
1	Production down, critical situation	Within 30 minutes
2	Major component of product disabled	Within 2 hours
3	Problem with the product, but customer has work-around solution	Within 4 hours
4	"How-to" questions and enhancement requests	Within 4 hours

## Product Support Policy

ASG fully supports the current release and one previous release of each of its products. ASG will temporarily support an older release, for up to six months, to provide time for you to upgrade.

After programming support for a product release is withdrawn, ASG will no longer supply new fixes for problems nor accept enhancement requests for that release. When a vendor announces the end of support for system software or a hardware configuration on which ASG products rely, ASG will make a similar announcement regarding the support plans for its products. ASG's support for problems affected by system software release levels will terminate when the vendor no longer supports their hardware or software.

Announcements regarding support plans for various products can be found on ASG's Web site.

## ASG Documentation/Product Enhancements

Use ASG's [contact](#) form to submit your product and documentation suggestions. Ensure that you include the name of the ASG product you are using.

# What's New?

This topic provides information on new features and services that are available in ASG-JOB/SCAN (herein called JOB/SCAN) and ASG-DOCU/TEXT (herein called DOCU/TEXT).

## ASG ACCESS Portal Updates and Service Packs

ASG's Intelligent Support Portal (ISP) has been changed to ASG Customer Care Expert Solution Services (ACCESS) Portal. We encourage you to visit ACCESS Portal: <https://access.asg.com> to verify whether any product or documentation revisions, new maintenance, or service packs apply to this product release.

### 8.0.1

This section lists the enhancements that are included in ASG-JOB/SCAN 8.0.1.

#### JOB/SCAN Plug-in

ASG-JOB/SCAN Plug-in is a Java plug-in that strategically integrates the mainframe capabilities of ASG-JOB/SCAN into Eclipse, IBM Explorer for z/OS and Topaz environments. The plug-in provides developers and production control staff an intuitive user interface to initiate JCL integrity, verification, and reporting. See [ASG-JOB/SCAN Plug-in for Eclipse IDE](#) for more information.

#### JOB/SCAN REST API

ASG-JOB/SCAN REST API interacts between JOB/SCAN Plug-in and JOB/SCAN REST server. This feature is required if you want to run the JOB/SCAN Plug-in. The plug-in also has a separate REST interface that you can use to automate tasks on the CI/CD environment. See [ASG-JOB/SCAN REST API Documentation](#) for more information.

### 8.0.0

This section lists the enhancements that are included in ASG-JOB/SCAN and ASG-DOCU/TEXT 8.0.0.

#### z/OS

### Functional Enhancements

JOB/SCAN 8.0.0 will run successfully on the current (z/OS 2.3) IBM mainframe hardware and operating system release.

It provides the below functional enhancements, which includes support for many of the new features in z/OS 2.3 JCL and utility programs:

- delimiters (DLM) parameter length from 2 to 18 characters.
- new JCL variables, SYSJOBNM symbol representing the JES2 job name and SYSJOBID symbol representing the JES2 job ID value.

- new DD parameter ROACCESS. JOB/SCAN processes ROACCESS sub-parameters such as ALLOW, DISALLOW, EXTLOCK and TRKLOCK.
- JCL NOTIFY statement to accept e-mail address, user-id, type of message and an expression. The syntax of NOTIFY statement:

```
NOTIFY EMAIL=address,USER=userid,TYPE=EMAIL|MSG,WHEN=expression
```

- EMAIL parameter to code an e-mail address in JCL for JOB and JOBGROUP statements.
- PASSWORD parameter in JCL for JOB and JOBGROUP statements.
- SYSEMAIL system symbol.
- GDGBIAS keyword and its sub-parameters on JOB statement.
- system variables with names longer than 8 characters.
- DD parameter SPIN and its sub-parameters.
- new DB2 12 commands:
  - FREE STABILIZED DYNAMIC QUERY (DSN)
  - BIND SERVICE (DSN)
  - FREE SERVICE (DSN)
- the keyword options that have been added to the following DB2 12 commands:
  - BIND PACKAGE (DSN)
  - REBIND PACKAGE (DSN)
  - FREE PACKAGE (DSN)
  - REBIND TRIGGER PACKAGE (DSN)

# Release Notes

This document describes the software changes that are included in Version 8.0.0 of ASG-JOB/SCAN (herein called JOB/SCAN) and ASG-DOCU/TEXT (herein called DOCU/TEXT). Before you install the product, review all attachments and follow instructions in [Installation](#).

## ASG ACCESS Portal Updates and Service Packs

ASG's Intelligent Support Portal (ISP) has been changed to ASG Customer Care Expert Solution Services (ACCESS) Portal.

We encourage you to visit ACCESS Portal: <https://access.asg.com> to verify whether any product or documentation revisions, new maintenance, or service packs apply to this product release.

## ACCESS Community User Forum

The new ASG online user forum - ACCESS Community, helps you to initiate real-time conversations with peers, share information, ask questions, receive latest announcements, and expertise on a variety of ASG product-based topics. To receive latest announcements on JCL products, follow the JCL Management topics.

### *To follow JCL Management topic:*

1. Visit the ACCESS Community Page at: <https://www.asg.com/en/Support.aspx>
2. Click 'COMMUNITY' from the list of topics.
3. Login using your ASG user credentials.
4. Accept the ACCESS Communities Rules of Engagement.
5. Select the 'OPERATIONS MANAGEMENT' from the list of Community Topics and click 'JCL Management'.
6. Click 'Follow'. You may choose to receive alerts about New posts and comments or New Posts only. A confirmation message is displayed at the top of the window.

### *To submit a new post:*

- Navigate to the JCL Management product topic, click the 'New post' button and describe the details regarding your post and click 'Submit'.



- For detail information on Access Communities, see the instructional video - <https://vimeo.com/313883669/a1072df43f>
- If you do not have access to ACCESS Communities or if you have questions about ACCESS Communities, please contact [support@asg.com](mailto:support@asg.com).

## Release Level

This document accompanies JOB/SCAN and DOCU/TEXT release level 8.0.0 and supersedes any prior release notes for these products.

## Operating System Requirements

JOB/SCAN and DOCU/TEXT 8.0.0 are supported on all IBM-supported releases of the z/OS operating system, including the most current release.

## Installation Notes

Starting with Version 8.0.0, the installation files and procedure for JOB/SCAN and DOCU/TEXT have changed significantly from previous releases.

- The product files are now delivered via download from theSG Customer Care Expert Solution Services (ACCESS) Portal in XMI file format instead of the ASG-POP package used in earlier releases.
- The product now provides full installation and maintenance support through IBM's System Modification Program/Extended (SMP/E).
- JOB/SCAN and DOCU/TEXT share the same product installation files, which are used to install both products. The DOCU/TEXT product files are included in the ZIP files and are transferred with the JOB/SCAN files to the mainframe. As a result of this consolidation, the DOCU/TEXT installation files have been renamed using this convention:

*JJS.nnn.\**

where *nnn* is the product version (base release level).

After you download these files and transfer them to your mainframe, you set up and customize the DOCU/TEXT environment separately from JOB/SCAN.

### **To install JOB/SCAN and/or DOCU/TEXT**

1. Download the 8.0.0 ZIP file from the ACCESS Portal: <https://access.asg.com> to your PC.
2. Extract the installation files from the ZIP file to your local directory.
3. Follow the appropriate procedure in [Installation](#) for the product(s) that you are installing.

If you have questions about the new library names and procedure or need assistance, contact ASG Customer Support.

## Release Library Update Summary

When upgrading from earlier versions of JOB/SCAN and/or DOCU/TEXT, replace all libraries.

### **Libraries Updated by Release**

These library changes have been implemented in release level 8.0.0.

- SJSSAMP members ENQDT and ENQJS are no longer being shipped.
- The .REXX library has been eliminated. As a result, RKIT and RSMP are now located in SJSSTD.L.
- .VBCLIST and .FBCLIST have been combined into SJSCLST.





Significant changes have been made to JOB/SCAN, so that it allows the JOB/SCAN Exits (Standards, Selection, and Program-name) to conform to a Language Environment (LE) and be language-independent. Any LE-conforming language that IBM supports, can be used as a JOB/SCAN exit.



- The COBOL Execution Environment (CXE) Run Option is deprecated. This means that CXE is no longer required or supported.
- Perform the listed changes related to Standards and Selection Exits only when you are upgrading JOB/SCAN from 7.8.1 or earlier version to 8.0.0 version. Changes are not required if you are upgrading from JOB/SCAN 7.9.0 version to 8.0.0 version.

## COBOL

All existing JOB/SCAN exits written in COBOL (Standards and Selection exits) must be compiled (or re-compiled) with an MVS Enterprise COBOL compiler. Enterprise COBOL version 5.1 and later are supported. An ISPF Edit Macro called CCEXIT is provided in JOB/SCAN's SJJSSAMP data set that can be used to compile any JOB/SCAN exit. Consult CCEXIT for changes that must be made to run at your site.

JOB/SCAN COBOL Standards and Selection exits must NOT issue an EXIT statement to return to JOB/SCAN. GOBACK must be used instead to return to JOB/SCAN.

For JOB/SCAN COBOL Standards exits, RETURN-CODE MUST be explicitly set to zero before returning to JOB/SCAN.

### For Example

```
MOVE ZERO TO RETURN-CODE.
```

```
GOBACK.
```

If RETURN-CODE is not explicitly set to ZERO, unpredictable results may occur.

## ASSEMBLER

Assembler exits must conform to an LE. This means that for any LE-compliant program, the entry macro CEEENTRY MAIN=NO and other CEE macros must be used. An ISPF edit macro called ASMEXIT is provided in JOB/SCAN's SJJSSAMP data set to assist in assembling JOB/SCAN exits written in assembler.

For JOB/SCAN Standards exits written in z/OS assembler, the exit MUST return a return code of 0 (zero) or unpredictable results may occur.

## REXX

Standards exits written in REXX require no changes.



All compiled exits must be link-edited AMODE(31). Enterprise compilers for COBOL version 5.1 and later default to AMODE(31). AMODE(31) is the default in the CCEXIT and ASMEXIT ISPF Edit Macros. See SJJSSAMP data set for details on CCEXIT and ASMEXIT.

# Glossary

This glossary defines product-specific terms used in JOB/SCAN documentation. The glossary does not contain industry- or IBM-standard terminology, which can be referenced in standard texts.

## #J statements

CA-7 Scheduler control statements. These may need to be preprocessed by a selection exit. See also *selection exit*.

## Access Control statements

JOB/SCAN run control statements that define user access privileges to options on the ASG-JOB/SCAN Primary Menu and to JOB/SCAN panels at the userid and groupid level. These are specified in JSOPT01 in the SJJSPARM library.

## Administrator

See *Product Administrator*.

## advisory message

An error message with a severity level of zero. Advisory errors are informational only and are unlikely to affect processing.

## assigned options member

An options member assigned to a specific userid by the Product Administrator. A personal options member overrides an assigned options member. In turn, an assigned options member overrides a default options member. See also *default options member* and *personal options member*.

## batch mode

A mode of executing JOB/SCAN in a background environment. See also *panel-assisted batch* and *batch*.

## CA7J1 / CA7J2

Sample selection exits provided by ASG to read and evaluate CA-7 #J cards and cause JOB/SCAN to process the JCL accordingly. See also *selection exit*.

## call types

Calls that are made to the Standards Facility program must have a type; for example, JOB statement call, DD statement call, end-of-step call, etc.

## CCEXIT member

CCEXIT residing in SJJAMP used to compile and link edit JOB/SCAN Cobol exits.

## Control Tables

Tables of values that establish valid device codes, I/O codes, nonstandard PARMs, and other site-specific values. These are specified in SJJSPARM library members.

## CSMP

A sample COBOL standards program residing in the SJJSTDL library. CSMP contains a representative set of validation samples that can be used as models in creating user-specific standards programs.

## default options member

An options member (JSOPT01 through JSOPT03, and JSOPT04) delivered in the SJJSPARM library at installation. Assigned options members or personal options members can be coded by the Product Administrator and assigned to specific users to override default options members. See also *assigned options member* and *personal options member*.

## Detail-level control statements

Statements used to customize JOB/SCAN execution. These statements have specific formats and required parameters, and are included after the appropriate group-level control statements. They can be coded either in the SJJSPARM library members or in the SYSIN stream of a batch JOB/SCAN execution. See also *group-level control statements*.

## Edit macros

Facilities that invoke JOB/SCAN directly from ISPF Edit. See also JCHG, JEM, JMSG, JREF, JSCAN, JSUB, JVER.

## error

A JCL error assigned a severity level of 5 or higher by JOB/SCAN. These errors are likely to affect processing, and should be corrected before the JCL is submitted to MVS. If JOB/SCAN encounters a severity level of 20 or greater, processing is discontinued.

## Error Summary

A section of the Structured JCL Listing summarizing all syntax, hidden, run-time, and site-specific errors encountered during validation. This summary can accompany the Structured JCL Listing, or can appear by itself.

## extended error message text

A longer textual error message that can be included on the Structured JCL Listing. This longer message can also be called by the JMSG Edit macro when validating JCL using the JEM Edit macro. The message is contained in the DSSMSG00 member for JOB/SCAN messages. User-defined Standards program messages are stored in the DSSMSG90 member.

## Extended Parameter List

COPYLIB member residing in the SJJSTDL library that is used when compiling the standards program. The Extended Parameter List contains all data definitions used by JOB/SCAN when passing information to and from the Standards Facility. For COBOL programs, the member name is CEPL; for assembler programs, the name is AEPL.

## File Allocations statements

Statements used to allocate files dynamically during JOB/SCAN execution. The types of files allocated include PDSOUTs, work files, user PROCLIBs, etc. These statements are specified in the SJJSPARM library members.

## group code

The fourth character in a JOB/SCAN error message (example, DSS4192) that identifies the type of error message. A group code of 0 identifies internal errors; 1 through 8 identify syntax and run-time errors; 9 identifies Standards Facility error

messages.

### **groupid**

An ID determined from the MVS ACEE control block by default. A groupid can be specifically assigned with the Group ID Run Option. This ID is used in Access Control statements to determine each group's access to individual JOB/SCAN functions.

### **group-level control statements**

Used in SJJSPARM library members or in the SYSIN stream when executing JOB/SCAN in batch. These statements specify the category of detail-level statements that follow and include Run Options, Report Options, Input Selection, and Control Tables. See also *detail-level control statements*.

### **Hardcopy Output Characteristics panel**

Panel used to input the standard class, destination, and other print parameters for TSO hardcopy.

### **index number**

A unique number assigned to each JCL statement to identify the job, step number within a job, step number within a PROC, and file number within a step. The index number identifies each statement's relative position within the jobs currently selected for validation.

### **initialization**

The process performed when JOB/SCAN is invoked. The contents of JSOPT01 are processed, followed by the processing of JSOPT02 (or its replacement assigned or personal options member).

### **Input Selection statement**

The group-level run control statement that precedes certain detail-level statements that specify JCL libraries and select jobs to validate. These statements are used when running JOB/SCAN in batch mode.

### **ISPF batch**

See *panel-assisted batch*.

### **JCHG**

A JOB/SCAN Edit macro that invokes JCL Change to change or reformat JCL from within ISPF Edit.

### **JCL Detail Information**

The section of the Structured JCL Listing that contains JCL statements with symbolics and overrides applied. It can also include control cards, if desired. The error messages for the JCL errors detected are included just below the statements in error.

### **JCL Validation panel**

A panel that lets users specify the criteria necessary to execute JOB/SCAN in foreground, batch, etc. It also provides access to Reformatter and JCL Change.

## **JEM**

A JOB/SCAN Edit macro that initiates the validation process from ISPF Edit. After JOB/SCAN Validation is complete, the JCL is redisplayed in Edit mode with error messages embedded, so that corrections can be made immediately to the JCL.

### **JEM label**

A unique label associated with each statement that contains an error after JEM has been executed. These labels have the format Jxxx, where xxx can be any alphanumeric characters. When using JMSG to expand the text for specific error messages, these labels identify the error for which the extended text is desired.

## **JMSG**

An Edit macro used with the JEM Edit macro to display extended error message text for selected error messages.

### ***JOB Statement for Batch Submission panel***

Panel used to code the JOB statement used for JOB/SCAN panel-assisted batch.

## **JREF**

An Edit macro used to invoke Reformatter tables (but not commands) while in ISPF Edit on the JCL.

## **JS6PROC**

A PROC in the SJJSSAMP library invoked either by the JSMVS JCL, or by using execution option B (batch) from the JCL Validation panel. This PROC can be used for JCL Validation, JCL Change, and JCL Reformatter execution.

## **JSCAN**

An Edit macro that initiates the validation process directly from ISPF Edit, and displays the resulting Structured JCL Listing in browse mode.

## **JSMVS**

SJJSSAMP library member containing JCL to run JOB/SCAN in batch. JSMVS invokes JS6PROC.

## **JSUB**

An Edit macro that initiates the validation process directly from ISPF Edit. If errors are encountered, the Structured JCL Listing displays in browse mode. Users can then optionally submit the job for execution. If no errors are found, the job is submitted automatically and control returns to ISPF Edit.

## **JSVERIFY**

JCL in the SJJSSAMP library that assists in writing the Standards Facility program, by generating a listing of actual values that would be passed to the Standards Facility from JOB/SCAN.

## **JVER**

A JOB/SCAN Edit macro that initiates the validation process for JCL Change and Reformatter commands in ISPF Edit.

### **list selection**

The process of creating, modifying, and selecting lists of jobs for validation. Lists reside as members in a library and are composed of groups of jobs that are validated together.

### **lists**

A method of grouping frequently validated jobs under one member name. This name is used instead of the individual job names when selecting jobs for validation.

### **mask**

A wildcard character for selecting jobs to be processed by JOB/SCAN, Reformatter, or JCL Change. Also used in JCL Change commands.

### **MSGs**

The label displayed after JEM processing, which indicates the Error Summary lines as well as the actual error message lines.

### **NOTE**

The label displayed after JEM processing, which indicates statements that are not part of the current JCL, like cataloged PROCs and control cards. NOTE statements cannot be edited.

### **options member**

A generic term used to describe any PARMLIB member residing in the SJJSPARM library. The member names delivered at installation are JSOPT01 through JSOPT03, and JSOPTM. The Product Administrator can create other options members.

### **Options in Effect Report**

A JOB/SCAN report listing all run control statements for the current execution. The report is available in detail-level and summary-level versions.

### **panel-assisted batch**

The process of using JOB/SCAN ISPF panels to generate JCL to execute JOB/SCAN in batch.

### **SJJSPARM library**

The library provided by ASG at installation that contains the PARMLIB members used by JOB/SCAN to control processing. These can be customized for each site.

### **SJJSPARM member**

An options member residing in the SJJSPARM library. The options member names delivered at installation are JSOPT01 through JSOPT03, and JSOPTM. The Product Administrator can create other options members.

### **personal options member**

An options member created by the Product Administrator and stored in the SJJSPARM library that can be specified by users. Personal options members override any assigned or default options members users might otherwise use when running JOB/SCAN. To use a personal options member, the user must have access to the Assign Alternate Options Members panel. See [Administering JOB/SCAN](#) for more information about this panel.

## PIF - Program Interface

A program in the TWS environment that passes control to IBM's TWS to create a simulation of this scheduler function with JOB/SCAN.

### prefix

A way of selecting groups of jobs to be validated, changed, or reformatted. The keyword PREFIX= is used to specify the first few characters of the member names. For example, to validate all jobs that have GL as the first two characters of the member name, use PREFIX=GL.

### Processing Options panel

The panel used to specify some Run Options for JOB/SCAN. These parameters override similar parameters coded as Run Options in SJJSPARM library members.

### Product Administrator

The person(s) assigned to perform JOB/SCAN supervisory tasks using the Product Administrator panels. The tasks include coding run control statements, establishing special options members for different user groups, and displaying administrator information.

### pure batch

Executing JOB/SCAN from JCL members submitted directly to TSO, without the assistance of JOB/SCAN ISPF panels.

### range

A way of selecting jobs by specifying a range of member names. For example, to validate the members with names JOBA through JOBZ, enter JOBA-JOBZ in the selection field.

### Reformatter

A JOB/SCAN facility that realigns EXEC, PROC, and DD statement parameters so that the JCL is formatted consistently. Reformatter can also renumber the lines of the JCL and remove comment fields.

### Report Options statements

Run control statements that allow users to define and change aspects of the layout or information appearing on each report. A Report Options statement is followed by appropriate detail-level statements, and can be coded either in the SJJSPARM library members or in the SYSIN stream of a batch JOB/SCAN run.

### run control statements

Statements that establish site parameters and affect how information appears on output reports. JOB/SCAN uses seven types of run control statements: System Identification, Access Control, File Allocation, Control Tables, Run Options, Report Options, and Input Selection.

### Run Options statements

Run control statements that affect JOB/SCAN processing and define interfaces with other systems. They are coded in options members in the SJJSPARM library or in the SYSIN stream of a batch JOB/SCAN run.



**selection exit**

An exit that can preprocess the JCL before JOB/SCAN parses and validates it. See also *CA7J1* and *CAJ72*.

**severity level**

The “seriousness” of an error detected by JOB/SCAN. Severity level 0 (zero) indicates advisory messages; levels 1 through 4 indicate warning messages; levels 5 through 19 are error level messages; and severity level 20 and above is a catastrophic error causing processing to be suspended.

**SHOW OPTIONS**

An ISPF command that can be used on any JOB/SCAN panel to view the Run Options, Report Options and Control Tables in effect. (JOB/SCAN help panels are not a valid environment for this command.)

**SHOW REPORT**

An ISPF command that can be used on any JOB/SCAN panel to view only the Report Options. (JOB/SCAN help panels are not a valid environment for this command.)

**SHOW RUN**

An ISPF command that can be used on any JOB/SCAN panel to view only the Run Options. (JOB/SCAN help panels are not a valid environment for this command.)

**SHOW TABLES**

An ISPF command that can be used on any JOB/SCAN panel to view only the Control Tables in effect. (JOB/SCAN help panels are not a valid environment for this command.)

**SJSDemo library**

A library provided by ASG containing sample job streams for demonstrating or practicing JOB/SCAN functions.

**SJSRRPF library**

A library supplied by ASG containing RPFs and panels necessary to execute JOB/SCAN in the ROSCOE environment without ISPF.

**SJSRTSO library**

A library supplied by ASG containing RPFs necessary to execute JOB/SCAN in the ISPF ROSCOE/ETSO environment.

**SJSSAMP library**

A library provided by ASG on the release tape with members that contain:

- JCL and PROCs for batch execution.
- JCL pertaining to various external interfaces.
- Miscellaneous other items.

**SJL**

The acronym for Structured JCL Listing.

## Space Analysis

Summary information that appears after the JCL Detail Information and the Statistical Summary in the Structured JCL Listing. It is a composite list of all space requirements for new files created in a job, broken down by UNIT and VOLSER.

## standards enforcement

The term used to refer to site-specific error validation that can be performed by the Standards Facility. Site standards are coded in programs that interface with JOB/SCAN.

## Standards Facility

The JOB/SCAN facility that provides the ability to perform site-specific validation of JCL. This error validation is in addition to JOB/SCAN JCL Validation.

## standards program

The program used by the Standards Facility to perform site-specific validation of JCL during a JOB/SCAN run. Users modify the sample programs provided by ASG in the SJJSTDL library.

## Statistical Summary

Job statistics that appear after the JCL Detail Information section of the Structured JCL Listing. Statistics include the total number of PROCs, programs, DD statements, and records processed by JOB/SCAN. In addition, the Statistical Summary reports on the input and output tape DD statements analyzed by JOB/SCAN.

## SJJSTDL library

A product library provided by ASG with members that contain sample source code and COPYLIB members for standards programs.

## Structured JCL Listing (SJL)

A report JOB/SCAN creates as it analyzes jobs. The report documents all statements within a job in processing sequence. PROCs are expanded, symbolics resolved, overrides applied, and control cards included. Any detected errors are flagged with an error message immediately following the statement in error.

## suppressed errors

Errors detected by JOB/SCAN which have messages suppressed from the Structured JCL Listing by use of Control Table AW or the Suppress Error Messages field on the Processing Options panel (or in a SJJSPARM library member). When messages are suppressed, their total count is included in the Error Summary.

## table of call types

See *call types*.

## user function

An Access Control statement that can be included in JSOPT01 to designate access to site-specific functions from the ASG-JOB/SCAN Primary Menu. The Product Administrator can define a maximum of two functions.

**userid**

A parameter used by the Product Administrator to grant or deny access to various system functions by userid.

**user-supplied control tables**

Control Tables defined to pass dynamic site-specific information to the standards program. These tables might contain lists of valid system codes, accounting parameters, DSN prefixes, or any other information that is subject to frequent change.

**warning**

An error message with a severity level of 1 through 4. These errors may or may not affect processing, and should be examined to determine if they must be corrected before the JCL is submitted to MVS.

# Installation

This documentation provides instructions on how to install ASG-JOB/SCAN (herein called JOB/SCAN) and ASG-DOCU/TEXT (herein called DOCU/TEXT) in ISPF and batch environments. It describes the following functions:

- One-time tasks, such as downloading the product files and setting up the ISPF environment.
- Ongoing customizing tasks, such as modifying the product environment and allocating file storage.



To install DOCU/TEXT, you must first install JOB/SCAN and then set up and configure the DOCU/TEXT add-on components and environment.

Topic	Description
<a href="#">Installing the Product</a>	Provides step-by-step instructions for installing JOB/SCAN and/or DOCU/TEXT.
<a href="#">Creating the JOB/SCAN Environment</a>	Provides detailed steps and conceptual information for setting up the JOB/SCAN environment at a site.
<a href="#">Setting Up Optional Interfaces and Exits</a>	Describes how to set up optional JCL standards and interfaces.
<a href="#">Final Installation Tasks</a>	Explains how to allocate a work file, special considerations, and initializing JOB/SCAN.
<a href="#">Running the Product in an APF-authorized Environment</a>	Provides guidelines for running in an APF-authorized environment.
<a href="#">Enabling DOCU/TEXT</a>	Provides information and instructions on how to enable and allocate the functions provided in the DOCU/TEXT add-on component, such as “where-used” cross-reference queries, reports, User Text, and library management and graphics interfaces.
<a href="#">Calling JOB/SCAN from a User Program</a>	Provides additional information on invoking JOB/SCAN from a user program.

# Installing the Product

This topic explains how to install JOB/SCAN and/or DOCU/TEXT add-on component in both ISPF and batch environments. The installation has two phases:

- **Phase 1** of the installation process involves the one-time task of downloading the product files and setting up the ISPF environment.
  - **Phase 2** of the installation includes tasks for customizing the product for your site. During this phase, processing parameters and files defined in the product SJJSM LIB library members, called JSOPTs, will be modified.
- [Planning](#)
  - [Product Installation Options](#)
  - [Downloading and Transferring the Product Files](#)
  - [Installing the Product](#)
  - [Installing the TSO Online Facility](#)
  - [Adding JOB/SCAN to the ISPF Primary Menu](#)
  - [Allocating the Files](#)
  - [Applying Product Authorization Code\(s\)](#)

## Planning

Before you start the installation process, complete the planning process to ensure that the environment is ready for installation. This topic contains information that must be done before installing JOB/SCAN and DOCU/TEXT.

### ASG ACCESS Portal Updates and CUMMs

ASG's Intelligent Support Portal (ISP) has been changed to ASG Customer Care Expert Solution Services (ACCESS) Portal. We encourage you to visit ACCESS Portal: <https://access.asg.com> to verify whether any product or documentation revisions, new maintenance, or monthly CUMMs (monthly maintenance) apply to this product release.

### System Requirements

The system requirements for each data processing environment are listed below.

#### Mainframe Requirements

The following system software configuration is required for mainframe operation:

- z/OS Versions 1.10 through 2.4
- ISPF Versions 6.0 through 7.4
- DFSMS level V1R3 or higher
- Batch region size of 4096 KB or larger
- TSO logon region size of 4096 KB or larger below 16 MB
- Storage above the 16MB line for load modules and GETMAINS

## Subsystems

JOB/SCAN and DOCU/TEXT support the following subsystems:

- CICS Transaction Server (TS) Versions 4.1 through 5.5
- DB2 Versions 8.1 through 12.1
- IMS Versions 10.1 through 15.1

## Compilers / Languages

JOB/SCAN and DOCU/TEXT support these compilers and programming languages:

- Enterprise COBOL for z/OS Version 5.1 through 6.2
- High-level Assembler Releases 5.0 through 6.0



- Object-oriented extensions added to any of the COBOL compilers are not supported.
- IBM no longer supports older non-LE compilers when they are run in currently supported Enterprise environments.

## Product Installation Options

JOB/SCAN provides full installation and maintenance support through IBM's System Modification Program/Extended (SMP/E).

The product files contain the installation libraries and members for JOB/SCAN and the DOCU/TEXT add-on component. After you download the product files, these product installation options are available:

- JOB/SCAN only
- DOCU/TEXT only
- JOB/SCAN+DOCU/TEXT



If you are installing only DOCU/TEXT, complete the steps in [Downloading and Transferring the Product Files](#) and [Installing the Product](#), using the shared product libraries, and then proceed to [Enabling DOCU/TEXT](#), to set up and customize DOCU/TEXT.

## Downloading and Transferring the Product Files

This section explains how to download the files from ASG ACCESS Portal and transfer them to z/OS mainframe.

### Downloading the Product Files

The product installation files are contained in a ZIP file, available via download from the [ASG Customer Care Expert Solution Services \(ACCESS\)](#) Portal. After you obtain the installation files, you must transfer them to your z/OS mainframe.



JOB/SCAN and DOCU/TEXT share the same product installation files, which are used to install both products. After you download and transfer these files to your mainframe, you set up and customize the product environment according to the product(s) for which you are licensed.

**To download the product files to your PC**

1. Locate or create a local directory on your PC for the product installation files.
2. To access the installation ZIP file:
  - Download the ZIP file containing the product installation files from the [ASG Customer Care Expert Solution Services \(ACCESS\)](#) Portal to your PC.

`JJSnnn.zip`

where `nnn` is the product version (base release level).

3. Extract these installation files from the ZIP file to your local directory.

**Product Files**

These product files are release-level specific and required for all JOB/SCAN installations (including upgrades):

File Name	Description
JJS.R800A.F1.XMI	Product files
JJS.R800A.F2.XMI	Product files and LOAD modules
JJS.R800A.F3.XMI	Product files
JJS.R800A.RECEIVE	RECEIVE JCL to convert the .XMI product files to regular datasets
JJS.R800A.RECVSTAT	RECEIVE JCL to convert the static .XMI product files to regular datasets
JJS.R800A.SMPMCS.XMI	SMP/E installation dataset

**Static Product Files**

These product files do not change from release to release and are only required if you are installing JOB/SCAN for the first time. If you are upgrading from a previous release, they are optional.

File Name	Description
JJS.COBLIB.XMI	Static product files
JJS.A3800LIB.XMI	Static product files
JJS.AFPLIB.XMI	Static product files

File Name	Description
JJS.GRAFLIB.XMI	Static product files

### Transferring the Product Files

After you download the installation files, you must transfer them from your PC to your z/OS mainframe.

#### To transfer the installation files to the mainframe

- Pre-allocate the files to be transferred to the mainframe. Use your desired dataset *prefix*. For example:

	Cyls	%Used	XT	Device	DSORG	RECFM	LRECL	BLKSIZE
<i>prefix</i> .JJS.R800A.F1.XMI	10	100	1	3390	PS	FB	80	3120
<i>prefix</i> .JJS.R800A.F2.XMI	10	100	1	3390	PS	FB	80	3120
<i>prefix</i> .JJS.R800A.F3.XMI	1	100	1	3390	PS	FB	80	3120
<i>prefix</i> .JJS.R800A.RECEIVE	1	100	1	3390	PS	FB	80	3120
<i>prefix</i> .JJS.R800A.RECVSTAT	1	100	1	3390	PS	FB	80	3120
<i>prefix</i> .JJS.R800A.SMPMCS.XMI	1	100	1	3390	PS	FB	80	3120
<i>prefix</i> .JJS.COBLIB.XMI	10	100	1	3390	PS	FB	80	3120
<i>prefix</i> .JJS.A3800LIB.XMI	10	100	1	3390	PS	FB	80	3120
<i>prefix</i> .JJS.AFPLIB.XMI	1	100	1	3390	PS	FB	80	3120
<i>prefix</i> .JJS.GRAFLIB.XMI	1	100	1	3390	PS	FB	80	3120

Or

- Use FTP or TN3270 to upload the files to the mainframe.
  - From the local directory where the files are located, enter this command:

```
ftp mvshost
```

where *mvshost* is the TCP/IP host name for your mainframe.

- At the User prompt, enter a valid TSO user ID for the z/OS system to which the files are to be transferred.
- At the Password prompt, enter the TSO password.
- Specify the directory where the file is to be uploaded:

```
cd 'prefix'
```

where *prefix* is the high-level qualifier under which the transferred datasets are to be allocated.

- Enter this command:

```
bin
```

- Enter this command to suppress the prompts asking you to confirm that the high-level qualifier for each installation file to be uploaded matches *prefix*:

```
prompt off
```

- Enter this command to specify the default dataset characteristics:



```
quote site recfm=fb lrecl=80 blksize=3120 cyl pri=1 sec=1
```

Typically, the default setting `pri=1` is sufficient to allocate all required datasets, but you can increase the PRI value to avoid multiple extents.

- h. Enter this command to copy the XMI files to the mainframe:

```
mput jjs.*
```

- i. Enter this command when the process is complete:

```
quit
```

This is a sample FTP session (including the commands entered by the operator performing the file transfer):

```

D:\JOBSCAN 8.0.0>ftp mvshost
Connected to mvshost.domain.com.
220-FTPD1 IBM FTP CS V1R11 at mvshost.domain.COM, 17:15:27 on 2015-09-22.
220 Connection will close if idle for more than 5 minutes.
User (mvshost.domain.com:(none)): userid
331 Send password please.
Password: password
230 userid is logged on. Working directory is "userid.".
ftp> cd 'prefix'
250 "prefix." is the working directory name prefix.
ftp> bin
200 Representation type is Image
ftp> prompt off
Interactive mode Off .
ftp> quote site recfm=fb lrecl=80 blksize=3120 cyl pri=1 sec=1
200 SITE command was accepted
ftp> mput jjs.*
200 Port request OK.
125 Storing data set prefix.JJS.R800A.F1.XMI
250 Transfer completed successfully.
ftp: 4563120 bytes sent in 75.91Seconds 65.16Kbytes/sec.
200 Port request OK.
125 Storing data set prefix.JJS.R800A.F2.XMI
250 Transfer completed successfully.
ftp: 4325837 bytes sent in 68.71Seconds 63.92Kbytes/sec.
200 Port request OK.
125 Storing data set prefix.JJS.R800A.F3.XMI
250 Transfer completed successfully.
ftp: 120640 bytes sent in 8.38Seconds 59.02Kbytes/sec.
200 Port request OK.
125 Storing data set prefix.JJS.R800A.RECEIVE
250 Transfer completed successfully.
ftp: 3120 bytes sent in 0.00Seconds 3120000.00Kbytes/sec.
125 Storing data set prefix.JJS.R800A.RECVSTAT
250 Transfer completed successfully.
ftp: 3120 bytes sent in 0.00Seconds 2980000.40Kbytes/sec.
200 Port request OK.
125 Storing data set prefix.JJS.R800A.SMPMCS.XMI
250 Transfer completed successfully.
ftp: 181840 bytes sent in 1.72Seconds 105.78Kbytes/sec.
200 Port request OK.
125 Storing data set prefix.JJS.COBLIB.XMI
250 Transfer completed successfully.
ftp: 4563120 bytes sent in 75.91Seconds 65.16Kbytes/sec.
200 Port request OK.
125 Storing data set prefix.JJS.A3800LIB.XMI
250 Transfer completed successfully.
ftp: 4325837 bytes sent in 68.71Seconds 63.92Kbytes/sec.
200 Port request OK.
125 Storing data set prefix.JJS.AFPLIB.XMI
250 Transfer completed successfully.
ftp: 120640 bytes sent in 8.38Seconds 59.02Kbytes/sec.
200 Port request OK.
125 Storing data set prefix.JJS.GRAFLIB.XMI
250 Transfer completed successfully.
ftp: 6760960 bytes sent in 109.21Seconds 61.91Kbytes/sec.
ftp> quit
221 Quit command received. Goodbye.

```

## Installing the Product

This section provides step-by-step instructions for installing the product after you transfer the installation files to your z/OS system.



Installing JOB/SCAN also installs DOCU/TEXT. If you are licensed for DOCU/TEXT, you must first complete the JOB/SCAN installation and configuration before setting up DOCU/TEXT.

### To install JOB/SCAN

1. Specify prefix values for the datasets.
  - a. Select a prefix value for the SMP/E and product datasets. These are the datasets required to install and run JOB/SCAN. The examples in this installation procedure use PROD.JJS800.
  - b. Select a prefix value for the SMP/E RELFILES. In this installation procedure, this prefix is referred to as *rfprefix*. SMP/E requires this prefix to be different from the prefix for the product files.

Dataset Type	Example
Product	PROD.JJS800
<b><i>Your product DSN prefix:</i></b>	
SMP/E RELFILES	PROD.JJS800.INSTALL
<b><i>Your SMP/E RELFILES DSN rfprefix:</i></b>	

2. Perform a TSO RECEIVE on the product installation datasets listed in [Product Files](#).
  - a. Customize and submit this dataset:

```
prefix.JJS.R800A.RECEIVE
```

where *prefix* is the user-defined high-level qualifier under which the transferred datasets have been allocated.

- b. In the JCL, change the high-level qualifier of the input datasets to be the same as the datasets you transferred. Specify the *rfprefix* you selected in step 1 as the high-level qualifier for the output datasets.

These files are created (where LJJS800 is the JOB/SCAN FMID):

File Name	Description
<i>rfprefix</i> .LJJS800.F1	80-byte, non-load module elements
<i>rfprefix</i> .LJJS800.F2	Load module elements
<i>rfprefix</i> .LJJS800.F3	Variable block elements
<i>rfprefix</i> .LJJS800.SMPMCS	SMP/E ++FUNCTION

3. Perform a TSO RECEIVE on the static installation datasets listed in [Static Product Files](#).

a. Customize and submit this dataset:

```
prefix.JJS.R800A.RECVSTAT
```

where *prefix* is the user-defined high-level qualifier under which the transferred datasets have been allocated.

b. In the JCL, change the high-level qualifier of the input datasets to be the same as the datasets you transferred.

Typically, the static datasets do not change for each release of JOB/SCAN. You can name the output datasets however you like.



This step is optional if you are upgrading from a previous release of JOB/SCAN.

These files are created:

File Name	Description
<i>prefix</i> .JJS.COBLIB	80-byte, non-load module elements
<i>prefix</i> .JJS.A3800LIB	IBM A3800 print files
<i>prefix</i> .JJS.AFPLIB	Non-A3800 print files
<i>prefix</i> .JJS.GRAFLIB	Page mode graphics and fonts files

4. Allocate the SMP/E datasets and control files.

a. Use member JSDEFCSI in *rfprefix*.LJJS800.F1 to allocate and define the SMP/E CSI dataset as indicated in the table

JOB/SCAN SMP/E CSI Dataset

DD Name	Library Type	Example DSN
SMPCSI	SMP CSI	PROD.JJS800.SMPCSI.CSI

***Your install DSN:***

b. Use member JSSMPAL in *rfprefix*.LJJS800.F1 to allocate the remaining SMP/E control files indicated in the table. Use the product prefix you selected in [step 1](#) for the high-level qualifier for these datasets.



The product prefix in the examples is PROD.JJS800. You can record the dataset names that you use on the following pages and print the pages for future reference.

### JOB/SCAN SMP/E Control Files

DD Name	Library Type	Example DSN
SMPLTS	SMP/E control files	PROD.JJS800.SMPLTS
	<i>Your install DSN:</i>	
SMPPTS	SMP/E control files	PROD.JJS800.SMPPTS
	<i>Your install DSN:</i>	
SMPLOGG	Optional. SMP/E control files (global)	PROD.JJS800.SMPLOG.G
	<i>Your install DSN:</i>	
SMPLOGAG	Optional. SMP/E control files (global)	PROD.JJS800.SMPLOGA.G
	<i>Your install DSN:</i>	
SMPMTS	SMP/E control files	PROD.JJS800.SMPMTS
	<i>Your install DSN:</i>	
SMPSCDS	SMP/E control files	PROD.JJS800.SMPSCDS
	<i>Your install DSN:</i>	
SMPSTS	SMP/E control files	PROD.JJS800.SMPSTS
	<i>Your install DSN:</i>	
SMPLOGT	Optional. SMP/E control files (target)	PROD.JJS800.SMPLOG.T
	<i>Your install DSN:</i>	
SMPLOGAT	Optional. SMP/E control files (target)	PROD.JJS800.SMPLOGA.T
	<i>Your install DSN:</i>	
SMPLOGD	Optional. SMP/E control files (distribution)	PROD.JJS800.SMPLOG.D

DD Name	Library Type	Example DSN
		<i>Your install DSN:</i>
SMPLOGAD	Optional. SMP/E control files (distribution)	PROD.JJS800.SMPLOGA.D
		<i>Your install DSN:</i>

- c. Use member JSALLOC in *rfprefix*.LJJS800.F1 to allocate the JOB/SCAN target and distribution libraries indicated in the tables. Use the product prefix you selected in [step 1](#) for the high-level qualifier for these datasets.



The product prefix in the examples is PROD.JJS800. You can record the dataset names that you use on the following pages and print the pages for future reference.

#### JOB/SCAN Target Libraries

DD Name	Library Type	Example DSN
SJJSCLST	CLIST	PROD.JJS800.SJJSCLST
		<i>Your install DSN:</i>
SJJSDBRM	DBRM	PROD.JJS800.SJJSDBRM
		<i>Your install DSN:</i>
SJJSDEMO	Demo	PROD.JJS800.SJJSDEMO
		<i>Your install DSN:</i>
SJJSINST	Installation	PROD.JJS800.SJJSINST
		<i>Your install DSN:</i>
SJJSMLIB	ISPF messages	PROD.JJS800.SJJSMLIB
		<i>Your install DSN:</i>
SJJSLOAD	Load	PROD.JJS800.SJJSLOAD
		<i>Your install DSN:</i>
SJJS Parm	PARM	PROD.JJS800.SJJS Parm

DD Name	Library Type	Example DSN
		<i>Your install DSN:</i>
SJJSPLIB	ISPF panels	PROD.JJS800.SJJSPLIB
		<i>Your install DSN:</i>
SJJSRRPF	ROSCOE	PROD.JJS800.SJJSRRPF
		<i>Your install DSN:</i>
SJJSRTSO	ROSETSO	PROD.JJS800.SJJSRTSO
		<i>Your install DSN:</i>
SJJSAMP	Sample	PROD.JJS800.SJJSAMP
		<i>Your install DSN:</i>
SJJSLIB	ISPF skeleton members	PROD.JJS800.SJJSLIB
		<i>Your install DSN:</i>
SJJSSTD	JCL standards facility	PROD.JJS800.SJJSSTD
		<i>Your install DSN:</i>

## JOB/SCAN Distribution Libraries

DD Name	Library Type	Example DSN
AJJSCLST	CLIST	PROD.JJS800.AJJSCLST
		<i>Your install DSN:</i>
AJJSDBRM	DBRM	PROD.JJS800.AJJSDBRM
		<i>Your install DSN:</i>
AJJSDEMO	Demo	PROD.JJS800.AJJSDEMO
		<i>Your install DSN:</i>
AJJSINST	Installation	PROD.JJS800.AJJSINST
		<i>Your install DSN:</i>

DD Name	Library Type	Example DSN
AJJSMLIB	ISPF messages	PROD.JJS800.AJJSMLIB
		<b><i>Your install DSN:</i></b>
AJJSMOD0	Load	PROD.JJS800.AJJSMOD0
		<b><i>Your install DSN:</i></b>
AJJS Parm	PARM	PROD.JJS800.AJJS Parm
		<b><i>Your install DSN:</i></b>
AJJSPLIB	ISPF panels	PROD.JJS800.AJJSPLIB
		<b><i>Your install DSN:</i></b>
AJJSRRPF	ROSCOE	PROD.JJS800.AJJSRRPF
		<b><i>Your install DSN:</i></b>
AJJSRTSO	ROSETSO	PROD.JJS800.AJJSRTSO
		<b><i>Your install DSN:</i></b>
AJJSAMP	Sample	PROD.JJS800.AJJSAMP
		<b><i>Your install DSN:</i></b>
AJJS LIB	ISPF skeleton members	PROD.JJS800.AJJS LIB
		<b><i>Your install DSN:</i></b>
AJJSSTD L	JCL standards facility	PROD.JJS800.AJJSSTD L
		<b><i>Your install DSN:</i></b>

5. Create the JOB/SCAN SMP/E procedure.
  - a. Copy member SMPJS from *rfprefix.LJS800.F1* to your procedure library as *SMPJJS80*.
  - b. Customize the new member as needed. You will use this procedure for all JOB/SCAN SMP/E maintenance.



6. Define SMP/E CSI entries. Use these members in *rfprefix.LJJS800.F1* to create the entries:

File Name	Description
JSENTG	Defines global zone CSI entries
JSENTT	Defines target zone CSI entries
JSENTD	Defines distribution zone CSI entries



- You must customize the DDDEFs for SCEELKED, SCEELIB, COBLIB, CSSLIB, and SDSNLOAD in job JSENTT to match the library names for your system. For example, SCEELKED (a ddname in the JCL for JSENTT) has a default library name of CEE.SCEELKED. If the LE library name is different than the default (that is, if you installed the Language Environment under a different high-level qualifier), you must customize the SCEELKED DDDEF statement to point to that library.
- Additionally, as part of the customization, you must make these changes:
  - Edit JSENTT to change `*OHLQ*.JJS.VSCOBOL.COBLIB` to `*OHLQ*.JJS.COBLIB`.
  - Edit the ADD DDDEF (SYSLIB) statement in JSENTT to refer to `SJJSINST` instead of `SJJSINS0`.
  - Edit the ADD DDDEF (SYSLIB) statement in JSENTD to refer to `AJJSINST` instead of `AJJSINS0`.
- If you *do not* have DB2 installed at your site, you must edit the ADD DDDEF (SDSNLOAD) statement in JSENTT to substitute `SYS1.LINKLIB` for the DB2 library.

7. Define DDDEFs for the JOB/SCAN target and distribution libraries. Use member JSDDDEF in *rfprefix.LJJS800.F1* to define the DDDEFs.

8. Receive the JOB/SCAN function. Modify JSRECV in *rfprefix.LJJS800.F1*.

- a. Change SMPPTFIN to point to this dataset:

```
rfprefix.LJJS800.SMPMCS
```

- b. Change the RFPREFIX keyword on the RECEIVE SYSIN statement to specify the same *rfprefix* used in [step 1](#).

```
RECEIVE SYSMODS RFPREFIX(rfprefix)
```

This identifies the *rfprefix.LJJS800.F1*, *rfprefix.LJJS800.F2*, and *rfprefix.LJJS800.F3* files that were created. For example:

```
//SMPCNTL DD *
SET BDY(GLOBAL) .
RECEIVE SYSMODS RFPREFIX(PROD.JJS800.INSTALL) .
```

9. Apply the JOB/SCAN function. Use member JSAPPLY from *rfprefix.LJJS800.F1*.



A return code of 4 is acceptable for this job.

10. Accept the JOB/SCAN function. Use member JSACCEPT from *rfprefix.LJJS800.F1*.



- A return code of 4 is acceptable for this job.
- The JOB/SCAN component must be accepted to allow future maintenance levels to be applied.

11. Make the JOB/SCAN libraries available from the ISPF Primary Option Menu. See [Installing the TSO Online Facility](#) and [Adding JOB/SCAN to the ISPF Primary Menu](#).
12. *If you are installing JOB/SCAN for the first time or on a different CPU,* contact [ASG Customer Support](#) to obtain a product operating license key. See [Applying Product Authorization Code\(s\)](#) for more information.

## Installing the TSO Online Facility

There are three ways to make the JOB/SCAN libraries available from the ISPF Primary Option Menu:

- **Method 1.** Modify the existing TSO logon PROC. Use this method unless the logon PROC is not easily accessible.
- **Method 2.** Modify ALLOCJS in the SJJSCST library. Use this method as a temporary measure until a new logon PROC can be created.
- **Method 3.** This method consists of two parts:
  - Part A—Modify LIBDEFJS in the SJJSCST library.
  - Part B—Modify the JOB/SCAN SJJSCST edit macros to use LIBDEF allocations.
 Use both parts of this method if the JOB/SCAN libraries are allocated using the ISPF LIBDEF service.

### Method 1—Modify the TSO Logon PROC

#### *To modify the existing TSO logon PROC*

1. Concatenate (ahead of any other datasets) the JOB/SCAN datasets for these ddnames:
  - SYSPROC (CLISTS)
  - ISPLLIB (load library—may be replaced with a STEPLIB)
  - ISPPLIB (ISPF panels)
  - ISPMLIB (ISPF messages)

If the LE runtime libraries CEE.SCEERUN and CEE.SCEERUN2 are not in your LINKLIST, add them to ISPPLIB (or STEPLIB).

If the BLKSIZE parameter is not coded for each ddname, ensure that the first dataset in the concatenation has a block size equal to or greater than the largest dataset in the group. Code a BLKSIZE parameter for the first dataset.



Ensure that no member names are duplicated among the JOB/SCAN datasets, IBM-provided datasets, or site-specific datasets.

2. If ALLOCJS of LIBDEFJS is not used, allocate the required DD name, JLRCL, to your SJJSPARM library. For example:

```
//JLRCL DD DISP=SHR.DSN=X.X.SJJSPARM
```

It is not necessary to allocate JLSSR to the terminal. It will be dynamically allocated if JOB/SCAN requires it.

3. Because the TSO logon PROC has been changed, log off the terminal session and log on again using the modified logon PROC.

### Method 2—Modify ALLOCJS CLIST



If you choose this method, you must execute the CLIST before entering ISPF each time you run JOB/SCAN.

#### To modify member ALLOCJS in the SJJSCST library

1. In the PROC statement, change the Xs in parentheses after DJSNAME to the high-level qualifier used when loading the JOB/SCAN libraries. For example:

```
PROC 0 DJSNAME (ASG.JSnnn)
```

where *nnn* is the three-character JOB/SCAN version.

2. Add any other site-required ISPF library concatenations to the appropriate TSO ALLOC commands. (Refer to current logon PROCs.). If the LE runtime libraries CEE.SCEERUN and CEE.SCEERUN2 are not in your LINKLIST, add them to ISPLIB (or STEPLIB).
3. Verify that the JOB/SCAN datasets have the same or greater block size as other ISPF datasets in the concatenated sequences.
4. Because you modified the ALLOCJS member in the SJJSCST library, return to TSO READY mode and enter this command to execute the CLIST:

```
EXEC 'ASG.JSnnn.SJJSCST(ALLOCJS)'
```

where *ASG.JSnnn* is an example of a site-specific high-level qualifier, and *nnn* is the three-character JOB/SCAN version.

### Method 3 Part A—Modify LIBDEFJS CLIST

If your site allocates the JOB/SCAN libraries using the ISPF LIBDEF service, modify the LIBDEFJS member in the SJJSCST library.

1. Change the value of DSJNAME to the appropriate JOB/SCAN dataset prefix. For example:

```
DSJNAME = "ASG.JSnnn"
```

where *ASG.JSnnn* is an example of a site-specific high-level qualifier, and *nnn* is the three-character JOB/SCAN version.

2. If the LE runtime libraries CEE.SCEERUN and CEE.SCEERUN2 are not in your LINKLIST, add them to ISPLIB (or STEPLIB).
3. If it is possible for JOB/SCAN to be used from other programs (particularly in Edit macro mode) include the STACK command on the LIBDEF statements.
4. Continue with Part B to include ISPF LIBDEF commands within the JOB/SCAN Edit Macros members in the SJJSLST library.

The LIBDEFJS member in the SJJSLST library is automatically executed each time that you select the JOB/SCAN option from the ISPF Primary Option Menu.



- ISPF LIBDEF allocation service will temporarily add the JOB/SCAN libraries to the current concatenations. LIBDEFJS will remove the JOB/SCAN libraries from the concatenations when you exit JOB/SCAN.
- When using LIBDEFs to allocate the JOB/SCAN libraries, you must link the standards module without the DYN option in order to statically link other subroutines, such as J06YTBL.

#### Modifying the LIBDEFJS member in the SJJSLST library

```
/* rexx */
DJSNAME = "ASG.JSnnn"
Parmlib = "'DJSNAME'.SJJSPARM'"
address TSO
/* Allocate JLLOD */
rc = LISTDSI("JLLOD" "FILE")
if rc <> 0 then do /* is JLLOD allocated? */
  "alloc f(JLLOD) da('DjsName'.SJJSLMOD) reu shr" /* no. Allocate it */
  lodalc = "Y"
end
```

#### Method 3 Part B—Modify the JOB/SCAN CLIST Edit Macros

You must modify these JOB/SCAN CLIST edit macros:

- JCHG
- JMSG
- JSCAN
- JEM
- JREF
- JVER

#### To modify the CLIST edit macros

Refer to the example shown in the figure.

1. In the PROC statement, change the Ns in parentheses after DJSNAME to the high-level qualifier used when loading the JOB/SCAN libraries. For example:

```
PROC 0 DJSNAME(ASG.JSnnn)
```

where *nnn* is the three-character JOB/SCAN version.

2. Do not change this line.

```
IF &DJSNAME = &STR(XXXXX.XXXXX) THEN +
```

3. Insert these ISPF LIBDEF commands into each Edit macro.



- If the JOB/SCAN load library is not in the linklist, then you must allocate JLLOD with the JOB/SCAN load library.
- If the JOB/SCAN load library is in the linklist, then you do not need to include the JOB/SCAN SJJSLMOD in the ISPLLIB.

---

```
ALLOC FI(JLLOD) DA('&DJSNAME..SJJSLMOD') SHR REU
IF &LASTCC ^= 0 THEN +
  WRITE UNABLE TO ALLOCATE JOB/SCAN LOAD LIBRARY TO JLLOD
ELSE+
  SET &LODALC = 'Y'

/*** THE FOLLOWING ALLOCATIONS SET UP JOB/SCAN ***/
/*** LIBRARIES USING LIBDEF STATEMENTS ***/

ISPEXEC LIBDEF ISPLLIB DATASET +
  ID('&DJSNAME..SJJSPPLIB') STACK

ISPEXEC LIBDEF ISPMLIB DATASET +
  ID('&DJSNAME..SJJSMMLIB') STACK

ISPEXEC LIBDEF ISPLLIB DATASET +
  ID('&DJSNAME..SJJSLMOD') STACK
```

---

For example:

**Modifying the JEM Edit macro with LIBDEF JS commands**

```

1.   PROC 0  DJSNAME (XXXXX.XXXXX)
      CONTROL END(ENDO) NOMSG NOFLUSH
/*CONTROL END(ENDO) MSG LIST CONLIST SYMLIST NOFLUSH /*S80725*/
      ISREDIT MACRO (PARMSTR)

2.   IF &DJSNAME = &STR(XXXXX.XXXXX) THEN +
      DO
          ISPEXEC SETMSG MSG(JM022R)          /*JTE30780*/
          GOTO ENDING
      ENDO

3.   ALLOC FI (JLLOD) DA('&DJSNAME..SJJS LMOD') SHR REU
3.   IF &LASTCC = 0 THEN +
3.       WRITE UNABLE TO ALLOCATE JOB/SCAN LOAD LIBRARY TO JLLOD
3.   ELSE+
3.       SET &LODALC = 'Y'

      /*** THE FOLLOWING ALLOCATIONS SET UP JOB/SCAN ***/
      /*** LIBRARIES USING LIBDEF STATEMENTS          ***/

3.   ISPEXEC LIBDEF ISPPLIB DATASET +
3.       ID('&DJSNAME..SJJSPLIB') STACK

3.   ISPEXEC LIBDEF ISPMLIB DATASET +
3.       ID('&DJSNAME..SJJSMLIB') STACK

3.   ISPEXEC LIBDEF ISPLLIB DATASET +
3.       ID('&DJSNAME..SJJS LMOD') STACK

/* *****      ALLOCATE RUN CONTROL LIBRARY      *****
ISPEXEC SELECT PGM(J00YCKAL) PARM(JLRCL)          /*JTE30780*/
IF &LASTCC NE 0 THEN +
    DO
        SET &RCLALC = 'N'                          /*813647 */
        ALLOC DDN(JLRCL) DSN('&DJSNAME..PARMLIB') SHR
    ENDO

/* *****      CALL THE JEM SUPERVISOR          *****
ISPEXEC SELECT PGM(J0AYEMU) PARM(PR(E) &PARMSTR) /*JTE30780*/

ENDING: +
3.   ISPEXEC LIBDEF ISPPLIB
3.   ISPEXEC LIBDEF ISPMLIB
3.   ISPEXEC LIBDEF ISPLLIB
3.   ISPEXEC LIBDEF ISPSLIB
3.   IF &LODALC = 'Y' THEN FREE F(JLLOD)
ISPEXEC VGET (ZSPLIT)
IF &STR(&ZSPLIT) = &STR(YES) THEN +
    EXIT
IF &RCLALC = 'N' THEN FREE DDN(JLRCL)
EXIT

```

4. Repeat Steps 1 - 3 for each Edit macro.

## Adding JOB/SCAN to the ISPF Primary Menu

You can add JOB/SCAN as an option of the ISPF Primary Menu.

**To add JOB/SCAN to the ISPF Primary Option Menu**

1. Locate the panel called ISR@PRIM in the existing IBM-supplied ISPLIB library.
2. Copy ISR@PRIM to the JOB/SCAN SJSPLIB library.
3. Add JOB/SCAN to the existing list of menu items in the ISR@PRIM panel:

```
% J +JOB/SCAN ISPF Support Environment
```

4. Change the menu command definition to add the product to the ISPF Primary Option Menu.

For LIBDEF allocation, insert this line:

```
J, 'CMD (LIBDEFJS) '
```

For non-LIBDEF allocation, insert this line:

```
J, 'PGM (J00YDMU) NEWAPPL (JS) '
```

**LIBDEF Example**

In this example, the JOB/SCAN ISPF support environment indicated by `J, 'CMD (LIBDEFJS) '` has been added to the ISPF Primary Option Menu.

**Change ISPF Primary Option Menu—LIBDEF method**

```
1, 'PGM (ISRBRO)      PARM (ISRBR001) '
2, 'PGM (ISREDIT)    PARM (P, ISREDM01) '
3, 'PANEL (ISRUTIL) '
4, 'PANEL (ISRFPA) '
5, 'PGM (ISRJB1)     PARM (ISRJPA) NOCHECK '
6, 'PGM (ISRPTC) '
7, 'PGM (ISRYXDR)    PARM (ISR)      NOCHECK '
8, 'PANEL (ISRLPRIM) '
9, 'PANEL (ZSDSFOP2) NEWAPPL (ISF) '
J, 'CMD (LIBDEFJS) '
T, 'PGM (ISPTUTOR)   PARM (ISR00000) '
', '
X, 'EXIT'
```

**Non-LIBDEF Example**

In this example, the JOB/SCAN ISPF support environment indicated by `J, 'PGM (J00YDMU) NEWAPPL (JS) '` has been added to the ISPF Primary Option Menu.







date, change the default value to a number ranging from 01 to 45. The number must be two characters.

2. Submit the job.

## Resolving problems with the Product Authorization Code

### *To resolve problems*

1. Check the return codes and messages from the assemble and link run.
2. If the job does not successfully complete:
  - a. Check SYSIN to ensure that the data was correctly entered.
  - b. Check SYSLIB and SYSLMOD DD statements to ensure that the correct library names were entered.
  - c. Resubmit the job.
3. Contact ASG Customer Support if the above steps do not resolve your problem.

# Creating the JOB/SCAN Environment

The instructions in this topic explain how to customize JOB/SCAN. As conditions and requirements change at your site, you can change the initial settings.

## JOB/SCAN Customization

Setting up the JOB/SCAN environment involves these tasks:

- [Step 1—Modify Detail Statements](#) - Modify detail statements to specify the TITLE1 value and the user ID of the product administrator.
- [Step 2—Specify PROCLIBs in JSOPT02](#) - Specify the PROCLIBs used in the installation.
- [Step 3—Modify Run and Report Options in JSOPT02](#) - Review and/or modify the values assigned to Run Options and Report Options.
- [Step 4—Modify Control Tables in JSOPT02](#) - Review and/or modify the values assigned to Control Tables.
- [Step 5—Modify the Batch PROC](#) - Modify the PROC that runs JOB/SCAN in batch mode.
- [Step 6—Install Edit Macros](#) - Install the edit macros provided in the SJJSCSLST library.
- [Step 7—Set Up the Product Usage Facility](#) - Create a Product Usage VSAM file for JLLPUF.

## Step 1—Modify Detail Statements

The JSOPT01 member in the SJJSPARM library contains control information. Update the TITLE1 value and set up access authorization for the product administrator.

### Changing the TITLE1 Value

The TITLE1 value is usually the your company name. This value appears on printed report headings.

#### *To change the TITLE1 value*

- a. Locate the System Identification group-level statement in JSOPT01 in the SJJSPARM library.
- b. Edit the alphanumeric field to the right of TITLE1. Enclose the name in single quotes. For example:

```
SYSTEM IDENTIFICATION
    TITLE1='up to 32 characters'
```

### Assigning Access to the Product Administrator

The Access Control statements specify access to online panels. This is where the product administrator's user ID is specified.



If JOB/SCAN will not be invoked under ISPF, delete or comment out the Access Control group-level statement and all Access Control detail-level statements.

**To assign Product Administrator access**

- a. Locate the Access Control group-level statement in JSOPT01 in the SJJSPARM library.
- b. Edit the ADMINISTRATOR=USERID detail-level statement to reflect the user ID of the product administrator for your site. For example:

```
ACCESS CONTROL
  ADMINISTRATOR=USERID=myid
```

## Step 2—Specify PROCLIBs in JSOPT02

The JSOPT02 member in the SJJSPARM library contains the File Allocations statements that enable JOB/SCAN to find the necessary datasets. This is where the site-specific PROCLIBs are specified. PROCLIBs are searched in the order that libraries are specified. To automatically generate PROCLIB File Allocations statements see [System Administrator Tasks](#).

**To modify File Allocations**

- a. Locate the File Allocations group-level statement. For example:

```
FILE ALLOCATIONS
  JLJPA =SHR, DSN=SYS1 . PROCLIB
        =SHR, DSN=SYS2 . PROCLIB
        =SHR, DSN=SYS3 . PROCLIB, UNIT=SYSDA, VOLSER=TEST01
        =SHR, DSN=SYS3 . PROCLIB, UNIT=SYSDA, VOLSER=PROD01
```

- b. Replace the dataset names with the site-specific PROCLIB names. The UNIT and VOLSER parameters are required only for uncataloged datasets.



Because the JLJPA PROCLIB TYPE defaults to PDS, if the PROCLIB type is PANVALET, LIBRARIAN, or LIBRARIAN-3, then code the JLJPA PROCLIB TYPE Run Option statement. See topic - [Run Options Group](#) in the [System Administrator Tasks](#) documentation for more information about this Run Option.

- c. Add or delete PROCLIBs as required.
- d. If the BLKSIZE parameter is not coded, ensure that the dataset with the largest block size is first in the concatenation. Alternatively, code the BLKSIZE parameter on the first dataset in the concatenation, specifying the largest block size as its value. For example:

```
JLJPA=SHR, DSN=TEST . PROCLIB, BLKSIZE=13200
```

Use the same concatenation sequence as the site JES2 or JES3 concatenation. See [File Allocations Group](#) for a list of valid parameters for the File Allocation statements.

### Rules for JES2 and JES3 Control Statements

Follow these guidelines for JES2 /\*JOBPARM and JES3 // \*MAIN statements in jobs to be processed.

- Each JES2 /\*JOBPARM PROCLIB= and JES3 // \*MAIN P= statement requires a unique JLJPA File Allocations statement.

- One-, two-, or three-character PROCLIB names and JES2 statements in the format PROCLIB=xxx can use the ddname JLJPAxxx, where xxx is the actual PROCLIB name. For example, for the PROCLIB library PROCLIB=B=XYZ, add the File Allocations statement JLJPXYZ.
- If the PROCLIB name is four to eight characters long, replace the name JLJPA with the actual PROCLIB name. For example, for PROC libraries PROCLIB=TEST1 and PROCLIB=PAYROLL, add the File Allocation statements for TEST1 and PAYROLL.

## Examples

These examples show sample File Allocations statements and job-related parameters.

### Example 1

This example contains the JLJPA File Allocation statements for the PROCLIBs shown in the second and third examples.

---

```
FILE ALLOCATIONS
  JLJPA   =SHR, DSN=SYS.PROCLIB          PROCLIBS
         =SHR, DSN=SYS1.MISCPROC
  JLJPAXY =SHR, DSN=OPS.TEST.PROCLIB
         =SHR, DSN=PC.TEST.PROCLIB
  PROC01  =SHR, DSN=USER.TEST.PROCLIB
         =SHR, DSN=APPS.DEVL.PROCLIB
```

---

### Example 2

This example includes the /\*JOBPARM PROCLIB=XY statement.

---

```
//PAYJOB1 JOB (TEST), 'PAYROLL', CLASS=T
/*JOBPARM PROCLIB=XY
//PAYP1   EXEC   PAYPROC1, UNIT=3380, DISP=SHR
```

---

### Example 3

This example includes the /\*JOBPARM PROCLIB=PROC01 statement.

---

```
//PAYJOB2 JOB (PROD), 'PAYROLL', CLASS=P
/*JOBPARM PROCLIB=PROC01
//STEP1   EXEC   PAYMAST, NODE=PROC, DISP=OLD
```

---

## Step 3—Modify Run and Report Options in JSOPT02

The JSOPT02 member in the SJJSPARM library contains Run Options, Report Options, and Control Tables (discussed in [Step 4—Modify Control Tables in JSOPT02](#)).

- **Run Options.** These statements control the way JOB/SCAN processes the JCL and how information is displayed in reports.
- **Report Options.** These statements control the content and layout of JOB/SCAN reports, as well as whether each report is to be printed or displayed.

See [Options](#) for complete information about Run and Report Options.

**To review Run Options and Report Options**

- Locate the Run Options and Report Options group-level statements in JSOPT02, and review the detail statements to determine whether the entries are appropriate for your site.

**To change the format in which dataset names are specified on panels**

- Specify one of these Run Options:

```
ISPF SPECIFICATIONS=DSNFORMAT=TSO
```

**Or**

```
ISPF SPECIFICATIONS=DSNFORMAT=NONTSO
```

The default value is TSO (TSO conventions). Under this setup, JOB/SCAN attaches the user prefix as the first node of any dataset name not enclosed in quotes.



When changing DSNFORMAT to NONTSO, always enter the fully qualified dataset name on the panels and do not use quotes.

## Step 4—Modify Control Tables in JSOPT02

Control Tables reflect the site environment. At this point in the installation process, you might want to change values in these tables:

- **Control Table AA.** This table associates UNIT names with device codes, which indicate the types of devices. Alter entries to override device codes and define unit values as appropriate.
- **Control Table AQ.** This table specifies sort and utility control card library names. Add to or change existing entries to control the content of the Structured JCL Listing as appropriate. If SYSIN is the ddname for these libraries, Control Table AQ changes are not required.

See [Control Tables](#) for complete information about control tables.

### Run Options for Control Tables AA and AQ

These Run Options are associated with Control Table AA:

Run Option	Description
Default Unit Names (DUN)	Assigns unit names to DASD, mass storage, reel tape, and cartridge tape devices for cataloged datasets in cases where the product runs on a system that does not have IBM module IEFEB4UV available. Control Table AA must contain the entries defined in this Run Option (or their default names).
Device Codes (DVC)	Allows up to three device codes to be specified for overriding or adding to existing devices codes.

Run Option	Description
Catalog Access (CAC)	Controls access to the system catalog and VTOCs (in the case of disk datasets). If this statement is set to <b>NO</b> , the product does not access the catalogs or VTOCs.

This Run Option is associated with Control Table AQ:

Run Option	Description
Access Control Cards (ACC)	Reads members within the jobstream that contain control cards and prints them in the Structured JCL Listing. If this option is set to <b>YES</b> , control cards are included in the reports. DB2 and IDCAMS control cards are included if the DB2 and IDCAMS parameters are turned on in the Program Product Interface Run Option.

### Control Table AA Values

This table describes the Control Table AA values:

Value	Device Code	Description
AADISK	D	DASD. Mandatory when Control Table AA is required.
AATAPE	T	Tape. Mandatory when Control Table AA is required.
AASYSDAV	M	Mass storage. Mandatory when Control Table AA is required.
AA3480	K	Cartridge tape. Mandatory when Control Table AA is required.
AA <sub>xxx</sub>	R	Report or printed output.
AA <sub>xxx</sub>	C	Cards.
AA <sub>xxx</sub>	G	Graphics or CRT.
AA <sub>xxx</sub>	V	Virtual I/O (VIO).
AA <sub>xxxx</sub>	U	Unknown device.
AA <sub>xxxx</sub>	X	Dummy (null device). A device code of <b>DUMMY</b> is assigned if the DD statement contains a <b>DUMMY</b> positional parameter or the parameter <b>DSNAME=NULLFILE</b> has been coded. A <b>DUMMY</b> device code is associated for DD statements that contain a <b>DDNAME=DD</b> parameter.

Value	Device Code	Description
AAxxxx	*	Up to nine user-defined device codes (that is, three tapes, three DASD, and three reports).  For DOCU/TEXT, if the Flowcharts report is to be produced, specify a Control Table AV entry to provide symbols for each user-defined entry.

### Modifying Control Table AA Values

If IBM's MVS module IEFEB4UV is available and the Run Options Catalog Access Control (CAC=YES,VTOC) is specified, JOB/SCAN uses the Eligible Device Table (EDT) to validate the UNIT parameter in the JCL and to select the device code. In this case, no alterations to Control Table AA are necessary.

You must alter entries in Control Table AA when performing these functions:

- Overriding the device code for a given unit, such as 3380.
- Defining valid unit values and device codes if CAC=NO or module IEFEB4UV is not available.

#### To change Control Table AA values

- Locate the Control Tables group-level statement in the JSOPT02 member in the SJJSPARM library.

The entries beginning with AA associate unit names with device codes. Use the existing entries as a pattern: The two A's are placed in columns 6 and 7; the device type, name, or unit address begins in column 8; and the device code is placed in column 21. For example:

```
-----+-----1-----+-----2-----+-----3-----+-----4
      AACART          K
```

- Delete Control Table AA entries that are not needed. Add other names used at your site.



If you delete entries (example, AADISK) and do not code unit names in the IDCAMS control statement, you will need to code the Default Unit Names (DUN) Run Option. See [Default Unit Names \(DUN\)](#) for additional information about this requirement.

### Modifying Control Table AQ Values

The Control Table AQ values specify which libraries contain control statements and should print in the Structured JCL Listing. If these libraries are always allocated with particular DD names that are specified in the AB Control Table (example, SYSIN), they do not need to be specified here.

Add or change existing entries in this format:

```
-----+-----1-----+-----2-----+-----3-----+-----4
CONTROL TABLES
  AQSYSQ.SJJSPARM
```



## Step 5—Modify the Batch PROC

If you are going to run JOB/SCAN in batch, then you must modify the JOB/SCAN PROC (JS6PROC) in the SJJSSAMP library to specify the appropriate SJJSLMOD library and SJJSPARM library members.



If the LE runtime libraries CEE.SCEERUN and CEE.SCEERUN2 are not in your LINKLIST, add them to STEPLIB.

### To insert high-level qualifiers for the SJJSLMOD library

- a. Locate the JS6PROC member in the SJJSSAMP library.
- b. Change the value in the DJSNAME with the first two nodes of the dataset names containing your JOB/SCAN libraries (that is, the high-level qualifiers used when LOADALL was submitted). For example:

```
//JS6PROC  PROC  DJSNAME='xxxx.xxxx'
```

If you followed standard conventions, node 1 is unique to your site and node 2 is equal to the TRLSE value.

- c. To enable users to run JOB/SCAN in batch, store JS6PROC in a PROCLIB that is accessible to all users.

### To execute JOB/SCAN as a submitted batch job

- Modify the JOB statement in the JSMVS member in the SJJSSAMP library as necessary. The PRM symbolic indicates JCL Change (C), Reformatter (R), or a validation Scan (S). Modify this symbolic as necessary. For example:

```
//JSSTEP1  EXEC  JS6PROC,PRM=CRS
```

This specification runs all three facilities in a single pass. See [Validating JCL in Batch](#) for more information about running JOB/SCAN in batch.

## Step 6—Install Edit Macros

This table lists the JOB/SCAN edit macros provided in the SJJSCST library.

Macro	Definition
JCHG	Executes JCL Change or Reformatter on the JCL being edited.
JEM	Validates the JCL. Executes JOB/SCAN on the JCL being edited in an ISPF Edit session. Returns to Edit mode displaying an error summary at the top of the screen.
JMSG	Used following JEM to obtain further information about JCL error messages.
JREF	Executes Reformatter on the JCL being edited in an ISPF Edit session.
JSCAN	Validates the JCL. Executes JOB/SCAN on the JCL being edited in an ISPF Edit session, and

Macro	Definition
	presents results in Browse or View mode.
JSESP	This macro is described in the <a href="#">User Information - JOB/SCAN with ESP</a> .
JTWS	This macro is described in the <a href="#">User Information - JOB/SCAN with TWS</a> .
JSUB	Validates the JCL and submits the job if no errors are detected. If JCL errors were found, presents a panel for further action.
JVER	Validates JCL Change and Reformatting commands being edited in an ISPF Edit session using "Bx" Control Table entries or system defaults for reformatting.

### To install the Edit macros

- a. Ensure that the product SJJSCST library is allocated in the SYSPROC concatenation.
- b. For each edit macro, modify the DJSNAME parameter on the PROC statement to specify the first and second dataset name qualifiers used for JOB/SCAN libraries. For example:

```
PROC 0 DJSNAME(DSSI.JS800)
```



Do not make any changes to the DJSNAME IF statement: IF &DJSNAME = &STR(XXXXX.XXXXX) THEN  
+

See [Validating JCL with Edit Macros](#) for more information about invoking the JOB/SCAN edit macros.

## Step 7—Set Up the Product Usage Facility

System-wide JOB/SCAN usage information and individual usage detail are automatically stored in the product usage repository (a VSAM dataset). The Product Usage Facility option on the ASG-JOB/SCAN Primary Menu enables you to view these reports online and to export the data. Additionally, you can set up JOB/SCAN to export usage data to a sequential file suitable for inclusion in spreadsheets and other programs that accept comma-delimited data.

### To create the Product Usage Facility VSAM file

- a. Determine this information for the VSAM file:
  - o A dataset name.
  - o The initial size.
- b. Edit the JSVSAM member in the SJJSSAMP library.
  - o Modify the Job statement to comply with site practices.
  - o Input the CLUSTER name.

- Insert the dataset name for the usage file DATA.
- Insert the dataset name for the usage file INDEX.

#### Sample VSAM file creation JCL for JLLPUFDD

```
//JSVSAM JOB (ACCT), 'PROGRAMMER NAME'
//*****
//* CREATE THE JOB/SCAN PRODUCT USAGE VSAM FILE TO BE USED TO *
//* STORE SYSTEM-WIDE AND INDIVIDUAL PRODUCT USAGE INFORMATION *
//* *
//* REF: SAMPLIB (JSVSAM) *
//* *
//* NOTE: MODIFY JOB STATEMENT AND ITEMS INDICATED BY ARROWS <== *
//*****
//STEP01 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//JSUSAGE DD DISP=OLD,UNIT=SYSDA <== VALID UNIT
//SYSIN DD *
DEFINE CLUSTER -
(NAME (DSSI.USAGE.FILE) /* <== CLUSTER NAME */ -
RECORDSIZE (4176 32761) -
INDEXED -
KEYS (32 0) -
SHAREOPTIONS (2) -
FILE (JSUSAGE)) -
DATA -
(NAME (DSSI.USAGE.FILE.DATA) /* <== DATA NAME */ -
CYLINDERS (1 1)) /* <== DESIRED SIZE */ -
INDEX -
(NAME (DSSI.USAGE.FILE.INDEX)) /* <== INDEX NAME */
```

- Submit the JCL to create the file.



Because Product Usage Facility records are written for each JOB/SCAN use, grant all users UPDATE access to this file.

#### To set up JOB/SCAN to export usage data

- Create a download dataset with these attributes:

#### Attributes—Export Sequential File

General Data	Current Allocation
Management class . . . : MCSTD	Allocated cylinders : 1
Storage class . . . . : SCBASE	Allocated extents . . : 1
Volume serial . . . . : DSSI04	
Device type . . . . . : 3390	
Data class . . . . . : **None**	Current Utilization
Organization . . . . : PS	Used cylinders . . . : 1
Record format . . . . : FB	Used extents . . . . : 1
Record length . . . . : 256	
Block size . . . . . : 2560	
1st extent cylinders: 1	
Secondary cylinders : 1	
Data set name type :	SMS Compressible . . : NO

#### To allocate the Product Usage Facility VSAM datasets

- Locate the JLLPUF DD statement in the JSOPT02 member in the SJJSPARM library. For example:

**Portion of JSOPT02 before modification for Product Usage Facility**

```

...
*****JLLMRF  =SHR, DSN=NODE1.MGMTRPT          JOB/SCAN USAGE DATASET
*****JLLPUF  =SHR, DSN=NODE1.MGMTRPT2        JOB/SCAN NEW USAGE
DATASET
*****JLLPUDLT =SHR, DSN=NODE1.MGMTDNL        JOB/SCAN USAGE DOWNLOAD
FILE
...

```

- b. If you are upgrading from a JOB/SCAN release prior to 6.2.6A, replace the leading asterisks on the JLLMRF DD statement with blank spaces and substitute the existing usage file dataset name for NODE1.MGMTRPT. For example:

**Portion of JSOPT02 after modification of the JLLMRF DD statement**

```

...
JLLMRF  =SHR, DSN=MYSITE.MGMTRPT          JOB/SCAN USAGE DATASET
...

```

If you are upgrading from 6.2.6A and later, skip this step.

- c. For all customers, locate the JLLPUF DD statement. Replace the leading asterisks with blank spaces and substitute the new usage file dataset name for NODE1.MGMTRPT2. Specify the same dataset name used to create the VSAM file. For example:

**Portion of JSOPT02 after modification of the JLLPUF DD statement**

```

...
JLLPUF  =SHR, DSN=MYSITE.MGMTRPT2        JOB/SCAN NEW USAGE
DATASET
...

```

- d. For all customers, locate the JLLPUDLT DD statement. Replace the leading asterisks with blank spaces and substitute the dataset name used to create the download file for NODE1.MGMTDNL. For example:

**Portion of JSOPT02 after modification of the JLLPUDLT DD statement**

```

...
JLLPUDLT =SHR, DSN=MYSITE.MGMTDNL        JOB/SCAN USAGE DOWNLOAD
FILE
...

```



Users with their own PARMLIB should use different dataset names.

- e. Re-invoke JOB/SCAN to enable the dataset allocations.

**To maintain the JOB/SCAN Product Usage File**

Apply the usual site procedures that govern production VSAM files to the Product Usage Facility VSAM file when performing these functions:

- Periodically making backups
- Reorganizing

- Checking file allocation space

See [Starting a New Usage Period](#) for information about closing the current period and starting a new period.



The Product Usage Facility file continues to grow as records are added for each new JOB/SCAN use. Product Usage Facility VSAM File allocation space should be carefully checked on a regular basis.

# Setting Up Optional Interfaces and Exits

This topic describes how to set up optional features in JOB/SCAN, including the JCL Standards Facility, Selection Exit, Security Exit, and Library Management interfaces.

- [General Considerations for User Exits](#)
- [User Exit Message Format](#)
- [JCL Standards Facility](#)
- [Selection Exit Interface](#)
- [Security Exit Interface](#)
- [Library Management Interfaces](#)
- [Database Interfaces](#)
- [ROSCOE Interface](#)
- [Other Interfaces](#)

## General Considerations for User Exits

Observe the guidelines in this topic when preparing the interface for a user exit (program name exit, selection exit, etc.).

### Special Notes

If JOB/SCAN is running APF authorized, any exit (such as Selection, Standards, or program name exit) that JOB/SCAN calls must also reside in an APF authorized library. Any module the exit calls must reside in an APF authorized library as well.

### Language Specific Notes

Significant changes have been made to JOB/SCAN, so that it allows the JOB/SCAN Exits (Standards, Selection, and Program-name) to conform to a Language Environment (LE) and be language-independent. Any LE-conforming language that IBM supports, can be used as a JOB/SCAN exit.



The COBOL Execution Environment (CXE) Run Option is deprecated. This means that CXE is no longer required or supported.

### COBOL

All existing JOB/SCAN exits written in COBOL (Standards and Selection exits) must be compiled (or re-compiled) with an MVS Enterprise COBOL compiler. Enterprise COBOL version 5.1 and later are supported. An ISPF Edit Macro called CCEXIT is provided in JOB/SCAN's SJJSSAMP data set that can be used to compile any JOB/SCAN exit. Consult CCEXIT for changes that must be made to run at your site.

JOB/SCAN COBOL Standards and Selection exits must NOT issue an EXIT statement to return to JOB/SCAN. GOBACK must be used instead to return to JOB/SCAN.

For JOB/SCAN COBOL Standards exits, RETURN-CODE MUST be explicitly set to zero before returning to JOB/SCAN.

#### For Example

```
MOVE ZERO TO RETURN-CODE.
GOBACK.
```

If RETURN-CODE is not explicitly set to ZERO, unpredictable results may occur.

### ASSEMBLER

Assembler exits must conform to an LE. This means that for any LE-compliant program, the entry macro CEEENTRY MAIN=NO and other CEE macros must be used. An ISPF edit macro called ASMEXIT is provided in JOB/SCAN's SJJSSAMP data set to assist in assembling JOB/SCAN exits written in assembler.

For JOB/SCAN Standards exits written in z/OS assembler, the exit MUST return a return code of 0 (zero) or unpredictable results may occur.

### REXX

Standards exits written in REXX require no changes.

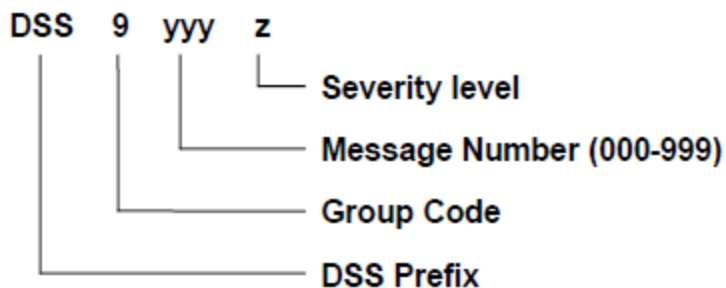


All compiled exits must be link-edited AMODE(31). Enterprise compilers for COBOL version 5.1 and later default to AMODE(31). AMODE(31) is the default in the CCEXIT and ASMEXIT ISPF Edit Macros. See SJJSSAMP data set for details on CCEXIT and ASMEXIT.

## User Exit Message Format

This section applies only to standards exits and program name exits. Selection exits cannot pass messages back to JOB/SCAN.

Messages passed back to JOB/SCAN from user exit programs follow the same format as other JOB/SCAN messages. A group code of 9 identifies a message as being from a user exit. The header fields in JOB/SCAN messages provide this information:



## Message Fields and Severity Level

This table describes the message fields:

Header Field	Description
Group Code	User exit messages are always identified with a Group Code of 9. This value is assigned by JOB/SCAN and cannot be changed by the exit.
Message Number	Three-digit numeric number that identifies the message as unique. The value is coded in the first three bytes of the message text by the exit routine. If no value is coded, JOB/SCAN uses 999 when displaying the message.
Security Indicator	This is based upon the severity code that the user exit passes back to JOB/SCAN. The severity indicator is translated as follows:

This table lists the security indicators for messages and their associated severity codes:

Severity Indicator	Severity Code
A—Advisory message	0
W—Warning message	1-4
E—Error message	5-19
C—Catastrophic	20-4096

## JCL Standards Facility

The JOB/SCAN JCL Standards Facility allows you to enforce site-specific JCL standards during the JCL Validation process. This section provides an overview of the steps required to enable this option.

### Activating the JCL Standards Facility

#### *To activate the JOB/SCAN Standards Facility*

1. Edit the JSOPT01 member in the SJJSPARM library. Add this line after the Administrator statement (under Access Control):

```
STANDARDS PROMPTER=USERID=xxxxxxxx
```

The USERID is typically the person responsible for defining standards. This statement limits access to the Standards Utility Maintenance panel functions.

2. Select **s** from the Standards Prompter panel. Specify the libraries that will be used by the Standards Facility. See the [System Administrator Tasks](#) for more information.



## Selection Exit Interface

A selection exit program is provided to pre-process JCL in situations where the JCL contains special non-MVS statements (example, for a scheduler or a reader exit). The JCL contained in a job can be automatically passed to this exit program, one line at a time, so that the program can add, change, or delete lines before JOB/SCAN processes it.

### Setting Up the Selection Exit

Selection Exit programs can be written in COBOL or other languages. Two sample COBOL Selection Exits are included in SJSSAMP library members CA7J1 and CA7J2. CA7J1 performs special processing for CA-7 #J cards. CA7J2 is an updated program, which performs CA-7 processing with multiple SCHED-IDs. See [Handling #J Cards with the CA-7 Scheduler Interface](#) and [Validating JCL that Includes CA-7 #J Statements](#) for information about handling #J cards.

Before JOB/SCAN validates the JCL, the lines of JCL are passed one at a time to the Selection Exit program. The exit then evaluates the line, makes changes if necessary, and requests an action based on the evaluation.

When all of the input JCL has been processed by JOB/SCAN, the Selection Exit is called with the CALL-SWITCH set to Z.

### To set up the Selection Exit

1. Copy the CA7J1 member or the CA7J2 member from the SJSSAMP library.
2. Modify the program to perform the functions necessary to meet your requirements. See [Selection Exit Return Codes](#) for information on setting return code values.
3. Compile and link the exit program to a library that is available to JOB/SCAN during run time.



To compile JOB/SCAN Cobol exits, use the ISPF edit macro CCEXIT found in SJSSAMP.

4. Enable the JOB/SCAN Selection Exit interface. These options are available:
  - Batch-only execution without panels
  - Foreground or panel-assisted batch execution

### Batch Execution

To enable the interface for batch execution without panels:

- Include a Selection Exit statement in the Input Selection section of the SYSIN stream before the first member name selected.

For example:

```
//SYSIN DD *
INPUT SELECTION
        SELECTION EXIT=SELPGM,' PARM1'
        SELECT      =PJGL0056
        DSN         =ASG.$PRD.SJJSDEMO
```

The Selection Exit is now automatically called for each JCL line.

### Foreground or Panel-assisted Batch Execution

To enable the Selection Exit for foreground or panel-assisted batch execution:

- Type the Selection Exit name (and parameter, if any) in the SELECTION EXIT and PARM fields of the ASG-JOB/SCAN - JCL Validation panel.

For example:

### Enabling the Selection Exit interface

```

----- JOB/SCAN - JCL VALIDATION -----
COMMAND ==>

      F - Foreground execution mode          P - Processing Options
      B - Batch execution mode              E - Edit JCL member
                                          BR - Browse JCL member

Input Library  ==> SYSUSR.JCL
Input Type    ==> J                      (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ==> PDS                    (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> LEDGR0                (Single member if BATCH with TRANSMIT)
Exclude Members ==>                      (Not valid for BATCH with TRANSMIT)
First Proclib ==>
Proclib Type  ==>                      (PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
Report Heading ==>
Display       ==> Y (Yes/No/ErrorsOnly)  Browse/View ==> B (B/V)
Hardcopy      ==> N (Yes/No/ErrorsOnly/SpaceOnly/JobErrors)
Options in Effect Listing ==> N (Y/N)      SJL PDSOUT ==> N (Y/N)

Selection Exit ==> SELPGM                Parm ==> PARM1

```

### Passing the Data Items

These fields, used for communication between JOB/SCAN and the Selection Exit, are included in the linkage section of the exit:

```

LINKAGE SECTION.
01  JCL-STATEMENT      PIC X(80) .
01  USER-PARM         PIC X(34) .
01  CALL-SWITCH        PIC X.
      88 FIRST-CALL-FOR-JOB          VALUE 'Y'.
      88 CALL-SWITCH-END-OF-ALL-MEMBERS VALUE 'Z'.
01  MEMBER-NAME        PIC X(10) .
01  USER-EXIT-WORD    PIC X(04) .

```

where:

- JCL-STATEMENT contains the 80-character JCL statement. Only the statements from the input JCL library are passed to the exit; cataloged PROC statements are not passed.
- USER-PARM is an optional user parameter that can be passed to the exit program in the USER-PARM field when the selection exit name is specified to JOB/SCAN.
- CALL-SWITCH is set to Y for the first call to the exit for each member. It is set to Z when all members have been processed. The exit program must reset this switch unless all members have been processed (CALL-SWITCH=Z).
- MEMBER-NAME contains the name of the member in which the current JCL statement resides.
- USER-EXIT-WORD is a word (4 bytes) which is for the user exits. This word is passed to all exits.

### Selection Exit Return Codes

The selection exit program generates return codes that determine how JOB/SCAN processes input statements. Use these

return codes to specify the action JOB/SCAN should take:

Return Code	Action to Be Taken
0	Process the line as is. The selection exit program has not changed this statement. For END-OF-ALL-MEMBERS call, no further processing is required.
4	Process the line as edited. The selection exit program has changed this statement.
8	Do not process this line.
12	Insert this new statement and call the selection exit again using the original statement (used to add data to the input stream).
16	Stop processing this member.
20	Terminate the JOB/SCAN run.

For example, to insert a line, place the line in the JCL-STATEMENT field and set the return code to 12.

### Running the Selection Exit

#### *To run a Selection Exit*

1. Determine what to evaluate on any JCL statement.
2. Code the logic necessary to perform the evaluation and any action that needs to be taken. For example, turn the line into a comment by placing `/**` in front of it.
3. Set the return code to the appropriate value to specify the action that you want JOB/SCAN to take.
4. Compile and link the sample Cobol exit. Use the ISPF edit macro CCEXIT found in SJJSSAMP.
5. Enable the JOB/SCAN selection exit interface by specifying the selection exit on the ASG-JOB/SCAN - JCL Validation panel (Foreground) or on an INPUT SELECTION statement (Batch).

## Security Exit Interface

The security exit interface is a user exit routine controlled by the Security Exit Run Option. It interfaces the product with the RACF, CA-ACF2, or CA-TOP SECRET security systems. This interface consists of three functions:

- Product access control
- DSNNAME validation
- Product termination



The security exit interface does not interfere with normal security logging procedures.

### Product Access Control

The product Access Control function interfaces with a security package (CA-ACF2/MVS, CA-TOP SECRET, or RACF) to verify

that the product job or TSO session has the proper access privileges to files prior to the actual access of those files. This prevents the product from causing security violations and possibly an S913 Abend.

While analyzing your JCL for syntax errors and to generate reports, the product might need to read the user catalog, the load libraries, control card libraries, and write to output files. The security exit is called prior to opening the file to verify that the user running the product has the proper access authority.

If access privilege is denied to a file, the exit returns a nonzero return code (in register 15) and no attempt is made to access the dataset. A message indicating that access has been denied is displayed on the ISPF panels or in the Structured JCL Listing (SJL).



The product run will continue. If you get an access denied message, note the file name and verify that you received the expected results.

### DSNAME Validation

DSNAME Validation validates the access to all datasets within the job stream being scanned by product. To use this function, code the DSNAME parameter on the Security Exit Run Option.

When DSNAME Validation is enabled, the product calls the Security Exit interface and passes it the USER or GROUP parameters found on the JOB statement or `//*LOGONID` statement in the job currently being scanned.

In this call, a blank dataset name is passed. If the USER or GROUP specified on the scanned job is different than the USER or GROUP of the product job or TSO session, then the Security Exit must be running APF-authorized in order to perform the DSNAME validation function. If you are running APF-authorized in foreground under TSO (in ISPF or ISPF Edit macro), link the Security Exit as RENT. The Security Exit name must be placed in table AUTHTSF in member IKJTSO00 of SYS1.PARMLIB. Check with a security administrator or systems programmer at your site.

If there is no USER or GROUP specified on the JOB statement of the job being scanned, the product omits this call. If the exit returns a nonzero return code (in register 15) from this call, this message is displayed in the Structured JCL Listing (SJL):

```
DSS4759A DSNAME SECURITY VALIDATION SUSPENDED FOR THIS JOB
```

DSNAME validation occurs after the product processes a DD statement from the JCL input stream that contains a dataset name. The expected access of Read, Update, or Alter is determined from the DISP parameter specified in the DD statement. This can be overridden by the I/O code (+I, +O) in the JCL comment field or from the Short Description file. If the exit returns a nonzero return code (in register 15) from this call, this message is displayed in the Structured JCL Listing (SJL):

```
          (READ  )
DSS4757E (UPDATE) ACCESS TO DATASET DENIED
          (ALTER )
```

The DSNAME validation function allows the user exit to pass a message back to the product to be displayed in the Structured JCL Listing. This message consists of a three-digit numeric value, which becomes part of the standard message number format and the text that makes up the short message. The length of this message cannot exceed 52 characters. A

severity code associated with the message is also passed back to the product. This severity code affects the return code from the current execution of the product and determines whether JOB/SCAN aborts processing.

Messages passed back to the product follow the same format as other product messages. See [User Exit Message Format](#).



If the SUPPRESS ERROR MESSAGES field on the JOB/SCAN Processing Options panel contains a value other than NO, the error suppression indicated by this field applies to error messages generated by the Security Exit as well as those generated by JOB/SCAN. See [The Processing Options Panel](#) for a detailed description of this and other fields on the Processing Options panel.

### Product Termination

Termination notifies the security exit that the product is terminating. This allows the security exit to free allocated storage and perform any termination routines.

### Security Exit Modules

Several security exit modules are provided to interface with various security systems. In addition to the load modules in the SJJSLMOD library, the source code for these programs is included in the SJJSSAMP library. Installations with special security exit requirements can modify these modules as required. See topic - [The Processing Options Panel](#) in the [Reference Information](#) for a detailed description of the programming interface.

This table lists the product security exit modules:

Module	Description
ACF2EXIT	This program interfaces with the CA-ACF2/MVS security package. It provides compatibility with security interfaces established for JOB/SCAN releases prior to 5.4. It does not support the dataset name validation function (DSNAME).
RACFEXIT	This program interfaces with the RACF and CA-TOP SECRET security packages. It provides compatibility with security interfaces established for JOB/SCAN releases prior to 5.4. It does not support the dataset name validation function (DSNAME).
SAF0	This program uses the System Authorization Facility (SAF) to interface with RACF, CA-ACF2/MVS, and CA-TOP SECRET. This program should not be used when running APF-authorized under ISPF with <code>DSNAME</code> coded in the Security Exit Run Option. It issues a RACINIT macro, and there is no way to specify a subpool for the ACEE.
SAF1	This program uses the third party RACHECK instead of the RACINIT/RACHECK used by SAF0. SAF1 uses subpool 251 for storage when invoked in an APF-authorized environment under ISPF. (This is because subpool 0 storage is deleted after the module is executed, while subpool 251 storage is not deleted until the job step task terminates.)
SAF2	SAF2 performs the same as SAF1 except that it uses the STATUS=ACCESS parameter of the RACROUTE macro to eliminate logging RACF violations when the user does not have proper access authority. CA-ACF2 requires APF authorization to use STATUS=ACCESS,

Module	Description
	unless a (CA-ACF2) SAFDEF record is defined with the NOAPFCHK keyword.

### Modifying a Security Exit Program

The security exit programs are included in the product files. You can modify them to accommodate your site-specific requirements. If you are running APF-authorized in foreground under TSO (in ISPF or ISPF edit macro), link the security exit as REENTRANT.

### Registers Used by the Exit Program

Each time the security interface is called, the registers are set as shown in this table:

Register	Contents
1	+0 Address of the Security Interface Parameter (SIP) list.
13	Address of a standard MVS register save area.
14	Return address.
15	Entry point address of the security interface exit routine.

### Program Requirements

#### *To ensure that the interface program functions properly*

1. After entering the exit routine, save all register contents.
2. Before returning control to the calling routine, restore all registers except Register 15, and ensure that Register 15 contains one of these return codes:
  - Zero—The product issues no messages.
  - Non-zero—The product issues message DSS4755 if the reason for the call was due to an access control function.

### Security Interface Parameter List

The Security Interface Parameter (SIP) list is passed to the security exit by the product. This table shows the contents of the SIP list.

Offsets Dec (Hex)	Type	Length	Name	Description
0 (00)	Character	4	SIPID	Control block ID (SIP).
4 (04)	Address	4	SIPSADDR	Virtual address of SIP.

Offsets Dec (Hex)	Type	Length	Name	Description
8 (08)	Character	1	--	Reserved field.
9 (09)	Character	11	SIPRSN	Reason the security exit is being called. These values are used for DOCU/TEXT access control: <ul style="list-style-type: none"> <li>• C 'CATALOG REA' —Read the user catalog.</li> <li>• D 'DYN ALLOC' —Allocate a library needed by DOCU/TEXT.</li> <li>• C 'EDIT REQST' —Edit the dataset using ISPF Edit.</li> <li>• C 'IMS LIBRARY' —Read dataset for IMS processing.</li> <li>• C 'MBR SELECT' —Read JCL dataset.</li> <li>• C 'READ CC MBR' —Read dataset for control card processing and PDS member verification.</li> <li>• C 'STEPLIB REA' —Read dataset for PGM verification.</li> <li>• C 'VTOC READ' —Read VTOC for dataset documentation.</li> <li>• C 'TERMINATION' —Processing has completed and DOCU/TEXT is terminating.</li> </ul>
20	Character	44	SIPDSN	Dataset name, which is to be used for validation. This field is blank for the termination call.
64 (40)	Character	6	SIPVOL	Volume serial number of the dataset, if available.
70 (46)	Character	1	SIPTYPE SIPREAD SIPUPDAT SIPALTER	Type of access desired: <ul style="list-style-type: none"> <li>• C 'R' —Read access.</li> <li>• C 'W' —Read and write access.</li> <li>• C 'A' —Alter access.</li> </ul>
71 (47)	Character	1	SIPDVTYP SIPDASD	Device type: <ul style="list-style-type: none"> <li>• C 'D' —DASD device.</li> </ul>

Offsets Dec (Hex)	Type	Length	Name	Description
			SIPTAPE	• C'T'—Tape device.
			SIPUNKWN	• C'U'—Unknown device.
72 (48)	Character	8	--	Reserved.
80 (50)	Character	8	--	Reserved.

## Library Management Interfaces

Optional interfaces are provided to enable the product to process JCL resident in CA-PANVALET library.

### CA-PANVALET Release 12 or Later

#### To enable the interface

- If the library that contains PAM (a CA-PANVALET load module) is not in the link pack area or in the link list, add it to the product load library concatenation list.
- If executing the product in batch mode, the concatenation list resides in JS6PROC.
- If executing the product in ISPF mode, modify the TSO logon PROC to add the library containing the PAM load module to the load library concatenation list in the PROC.

### CA-PANVALET Release 11 or Earlier

#### To enable the interface

1. Edit the LINKPAM member in the SJJSSAMP library.
  - a. Modify the JOB statement with your site-specific parameters.
  - b. Modify the DJSNAME symbolic to specify the first two nodes of the product load library name.
  - c. Modify the LIBNAME symbolic to specify the CA-PANVALET load library name.
2. Submit LINKPAM. The job should complete with a condition code of 00.

## Database Interfaces

Optional interfaces are provided that allow the product to process database information for IMS, DB2, and IMS/DB2. Follow the appropriate installation procedure.

Additionally, you can develop interfaces for other database packages using one of the data import features. Refer to [Reference Information](#) or [System Administrator Tasks](#) for more information.

### DB2 Interface

ASG provides a DB2 interface for extracting additional information for the product to process (including DB2 syntax, dataset references, etc.). The information is available for batch DB2 jobs executing the DSN RUN command using the



IKJEFT01, IKJEFT1A, and IKJEFT1B TSO driver programs. See [IMS/DB2 Interface](#) for information about jobs accessing DB2 databases through IMS.

### To enable the interface

1. Add the DB2 parameter to the Program Product Interface Run Option in the appropriate PARMLIB member. For example:

```
PROGRAM PRODUCT INTERFACE =DB2
```

2. Code the appropriate Control Table entries in the PARMLIB member. See [Control Table AB: Control Card DD Name](#) (related to JOB/SCAN) or [Control Table AB: Control Card ddnames](#) in ASG-DOCU/TEXT's System Administration guide (related to DOCU/TEXT) for details on Control Table AB. For example:

```
.....+.....1.....+.....2.....+.....3.....+.....4
ABSYSSTIN      IKJEFT01
ABSYSSTIN      IKJEFT1A
ABSYSSTIN      IKJEFT1B
```

The DB2 interface will be invoked automatically each time the product encounters a job that accesses a DB2 database through the DSN RUN command.

This is a sample batch DB2 job stream:

---

```
//STEP EXEC PGM=IKJEFT01
//      .
//      .
//SYSIN DD *
DSN  SYSTEM(DB2P)
RUN  PROGRAM(PAYDB)  LIB (' PROD.LOAD' )  PLAN (PAYDB)
END
```

---

### IMS Interface

ASG provides an IMS interface for extracting additional information for the product to process. The information is available for batch IMS jobs executing DFSRR00 programs through the DLI, BMP, and DBB functions. See [IMS/DB2 Interface](#) for information about jobs accessing DB2 databases via IMS.

### To establish the interface

1. Define the IMS ACB library in the File Allocation section of the appropriate PARMLIB member (BMP processing only). For example:

```
ACBLIB =DSN=PROD.IMS.ACBLIB,SHR
```



The product performs special processing for batch jobs that access IMS databases using the DFSRR00 driver with the DLI, BMP, and DBB options and the DFHDRP driver that accepts keyword parameters in the PARM field.

2. Add the IMS parameter to the Program Product Interface Run Option in the appropriate PARMLIB member. For example:

```
PROGRAM PRODUCT INTERFACE =IMS
```

3. Code the appropriate Control Table entries in the PARMLIB members. For example:

```

.....+.....1.....+.....2.....+.....3.....+.....4
      ADDFSRRC00      UR
      APDFSRRRC00      2

```

See [Control Tables](#) for details on Control Tables AD and AP for IMS.

The IMS interface is invoked automatically whenever the product encounters the IMS driver program accessing databases through the DLI, BMP, or DBB functions.

### IMS/DB2 Interface

ASG provides an IMS/DB2 interface for extracting additional information for the product to process. The information is available for batch IMS jobs executing DB2 programs via the DDITV02 DD control statements.

#### To establish the interface

1. Define the IMS ACB library in the File Allocation section of the appropriate PARMLIB member (BMP processing only). For example:

```
ACBLIB    =DSN=PROD.IMS.ACBLIB,SHR
```



The product performs special processing for batch jobs that access IMS databases using the DFSRRC00 driver with the DLI, BMP, and DBB options and the DFHDRP driver that accepts keyword parameters in the PARM field.

2. Add the IMS and DB2 parameters to the Program Product Interface Run Option in the appropriate PARMLIB member. For example:

```
PROGRAM PRODUCT INTERFACE    =IMS,DB2
```

Or

```
PROGRAM PRODUCT INTERFACE
      = (IMS,X16H) , DB2
```



X16H is a one- to four-character IMS subsystem ID (optional).

3. Code the appropriate Control Table entries in the PARMLIB members. For example, this table entry provides IMS with access to DB2 databases:

```

.....+.....1.....+.....2.....+.....3.....+.....4
      ABDDITV02      DFSRRC00
      ADDFSRRC00      UR
      APDFSRRRC00      2

```

See [Control Tables](#) (related to JOB/SCAN) or Control Table Entries in ASG-DOCU/TEXT's System Administration guide (related to DOCU/TEXT) for details about Control Tables AD and AP for IMS and Control Table AB for IMS access of DB2 databases.

The IMS interface is invoked automatically whenever the product encounters the IMS driver program accessing databases through the DLI, BMP, or DBB functions.

## ROSCOE Interface

ASG provides facilities that enable ROSCOE users to run JOB/SCAN in the ROSCOE RPF or ETSO environment. You can run the RPF and ETSO facilities independently or combined (that is, portions of the two facilities can be used simultaneously).

### RPF Installation Procedure

To install the ROSCOE RPF facility, you must complete these tasks:

- Execute member JSRUNLD.
- Modify member JSRMAIN.

### Step 1—Execute Member JSRUNLD

#### *To execute member JSRUNLD*

- a. Copy member JSRUNLD from the SJJSRRPF library and save it into a ROSCOE library suitable for RPF execution. JSRUNLD is an RPF, written to simplify the RPF and panel installation.
- b. Execute member JSRUNLD from the new location.
  - At the first prompt, enter the fully-qualified dataset name `&DJSNAME.SJJSRRPF` unloaded from the product files.
  - At the second prompt, enter the two- or three-character prefix of the ROSCOE library from which these RPFs will execute.
  - ASG recommends that you place these RPFs in a GLOBAL or system library that is accessible to all users.
  - Ensure that the ROSCOE ID has sufficient authority to update this library.

### Step 2—Modify Member JSRMAIN

#### *To modify member JSRMAIN*

- a. Attach member JSRMAIN and change the file name assigned to permanent variable P16 to correspond to the high-level qualifiers of the product libraries from the download (do not modify the last node).
- b. After making the necessary changes, resave the updated RPF.

### RPF Installation Panels

The following panels are typically displayed during the installation and initial execution of JOB/SCAN under the ROSCOE RPF.

```

                                JOB/SCAN RPF UNLOAD PANEL

Enter the data set name into which the JOB/SCAN RPFs were unloaded:
Enter the ROSCOE PREFIX into which the RPFs are to be copied:
===>


PRESS PF3/PF15 to terminate without RPF execution.
```

```

*****
*
*   This is the first execution of JOB/SCAN under your USERID.
*
*   JOB/SCAN performs many functions which require batch execution.
*   Other functions require selection of job JCL from JCL libraries.
*   The next two panels you will see require information which
*   JOB/SCAN will need to correctly perform these functions in a
*   ROSCOE environment.
*
*   When the entries are complete, the regular JOB/SCAN menu will be
*   presented.
*
*   Press ENTER to move on to the panels.
*   Press PF3/PF15 to exit and return to ROSCOE command mode.
*
*****
    
```

**ETSO Installation Procedure**

Prior to installing the ETSO facility, complete the regular product installation. If you do not intend to use the ISPF panels, the SJJSP LIB, SJJSM LIB, and SJJSC LST libraries are not required. The SJJSC LST library is not used in any case.

 The JOB/SCAN ISPF panels utilize PDF as well as ISPF. You must have the PDF extension of TSO/ISPF to take full advantage of the JOB/SCAN ISPF panels. However, it is not necessary to have TSO/ISPF to invoke JOB/SCAN and REFORMATTER interactively within ROSCOE.

To install the ROSCOE ETSO facility, you must complete these tasks:

- Establish the ETSO environment using member JSETSOPG or JSETSOP7.
- Modify member JSRISPF.
- Modify member JSCAN.
- Modify member JREF.

**Step 1—Establish the ETSO Environment**

**To establish the ETSO environment**

- a. Copy member JSETSOPG or JSETSOP7 from the xxxxx . xxxxx . SJJR TSO library and merge one of these entries into the ETSOPGMS member in your main ROSCOE library (usually the RO prefix).
- b. Verify that the entries are in alphanumeric sequence and replace the ETSOPGMS in the main ROSCOE library. This table lists the ETSOPGMS entries.

Entry	Function
ISPSTART	ISPF panels

Entry	Function
J00YDMU	JOB/SCAN panels
YCMUJ02YJS	ISPF panels

- c. If you plan to use the ISPF panels, you should install the member CLLEXIT (ACF2 Exit) into your ROSCOE.SOURCE library. This allows ROSCOE to return a usable user ID for an ISPF dialog. (CA/ROSCOE support can assist you further.)
- d. If your site security package is ACF2, you must define all of the program name entries in ETSOPGMS into the ACF\$RCLL member in ACF2.MACLIB (site dependent). You should assemble this member prior to any ETSO execution. This enables ROSCOE to execute the programs referenced in ETSOPGMS interactively.
- e. Add these ROSCOE initialization parameters to your ROSCOE startup parms:

Parameter	Description
ESCAPE=.	Escape sequence.
ETSALLOC=45	Max dynamic file allocation.
ETSOPGMS=RO.ETSOPGMS	Eligible program list.
ETSSLICE=3000	Time slice default.
ETSSRCH=ALL	Optional. Called PGMs (ETSOLIB, JOBLIB, LPA, and LLA). If you specify ETSSRCH=ALL, you can add your JOB/SCAN load library to the Link Pack Area (LPA), Link List Area (LLA), JOBLIB, STEPLIB, or ETSO DD statement. If you specify ETSSRCH=DD, you must concatenate the product load library in the ETSO library DD statement. Refer to the <i>ROSCOE System Reference</i> for ETSO application installation details.
PSCBUSER=aaaa	Pseudo UID.

## Step 2—Modify Member JSRISPF


Perform these steps only if you plan to use the ISPF panels via ROSCOE.

### To modify member JSRISPF

- a. Copy member JSRISPF from the `xxxxxx.xxxxxx.SJJSRTSO` library. Modify line 3. Change `xxxxxx.xxxxxx` to the high-level qualifier used when downloading the product files.

- b. Locate the line containing `ALLOC ISPPROF`. Change the `YOUR.` dataset name to reflect a naming convention compatible with your site standards.

The dataset should use a high-level qualifier that is accessible to all users for create and update functions. ASG recommends that you set the second node to your ROSCOE prefix. The dataset must be a PDS with the attributes of `RECFM=FB` and `LRECL=80`. Five tracks should be adequate.

 If the JSRISPF RPF does not build the dataset for you, it must be pre-allocated by/for each user prior to execution.

- c. Locate the lines that contain the string `YOUR.` and modify each one to reflect the appropriate IBM shipped ISPF/PDF library. It is not necessary, nor desirable, to identify any other vendor libraries.
- d. Save JSRISPF into a universally accessible ROSCOE library as `JOBSCAN` or any other desirable name.

### Step 3—Modify Member JSCAN

#### To modify member JSCAN

- a. Copy member JSCAN from the SJJSRSTO library. Modify line 3. Change the string `XXXXXX.XXXXXX` to the high-level qualifier used when downloading the product files.


You might need to locate the *lines* containing the string `+PREFIX+.WORK.JLSCRN` or `+KEY+.JLSCRN.WORK` and modify the dataset name to conform to your site standards.

- b. Save JSCAN into a universally accessible ROSCOE library as `JSCAN` or any other desired name.

### Step 4—Modify Member JREF

#### To modify member JREF

- a. Copy member JREF from the SJJSRSTO library. Modify line 3. Change the string `XXXXXX.XXXXXX` to the high-level qualifier used when downloading the product files.
- b. Save JREF into a universally accessible ROSCOE library as `JREF` or any other desired name.

 When attempting to execute any of these facilities under a ROSCOE release prior to 5.5, you must modify the JSCAN RPF and JSOPT02. ROSCOE releases prior to 5.5 do not support SVC 99 for dynamic allocation of concatenated datasets.

For PROCLIB allocation, the FILE ALLOCATION statement that will be affected is JLJPA. Additional statements might be affected as well, and you will need to add these allocations to the appropriate RPF.

The ROSCOE installation is now complete.

## Other Interfaces

Several other optional interfaces are provided that allow JOB/SCAN to validate scheduler package and tape management system information.

- For TWS, refer to [User Information - JOB/SCAN with TWS](#).
- For ESP, refer to [User Information - JOB/SCAN with ESP](#).
- For all other interfaces, refer to [Other Interfaces](#) in [Reference Information](#).

# Final Installation Tasks

This topic describes about the optional installation tasks and details the procedure for initializing and using JOB/SCAN.

- [Special Considerations](#)
- [Initializing JOB/SCAN](#)

## Special Considerations

After the required installation tasks are performed, determine if any of these optional installation tasks are required:

### **Modify the JCL for compiling and linking JOB/SCAN standards programs**

When writing a program to check site standards during a JOB/SCAN run, refer to [JCL Standards Facility](#) for information on establishing this option.

### **Setting up Optional Interfaces**

When scanning JCL stored in library management systems (CA-PANVALET library), or to take advantage of IMS and DB2 interfaces, see [Setting Up Optional Interfaces and Exits](#).

### **Setup JOB/SCAN to run APF-authorized**

JOB/SCAN provides some features for which IBM requires APF-authorization. These items include:

- Verification of JES2 and JES3 remote destinations prior to z/OS release V1R9.
- SMS information and validation.
- The ability to verify dataset access for a userid other than the one running JOB/SCAN. Refer to the DSNAME parameter in [Security Exit](#) run option.

See [Running the Product in an APF-authorized Environment](#) for further information and installation instructions.

## Initializing JOB/SCAN

The first time JOB/SCAN is accessed by a particular userid, the system steps through a sequence of four panels:

- JOB/SCAN Welcome panel.
- Hardcopy Output Characteristics panel.
- Job Statement for Batch Submission panel.
- Processing Options panel.

All the initialization panels are described below.

### ***To begin using JOB/SCAN***

1. Select the JOB/SCAN option from the ISPF Primary Option Menu.

The JOB/SCAN Welcome panel appears.



2. Press Enter to display the Hardcopy Output Characteristics panel.
3. Complete the Hardcopy Output Characteristics panel, then press End to continue with the initialization sequence.

### Changing Panel Information

The information entered in the initialization panels is saved from session to session in the ISPF profile. For example, the output class for hardcopy output designated in the Hardcopy Output Characteristics panel is always used when printing JOB/SCAN reports.

Table explains how to change information in these panels.

Changing panel information stored in ISPF profile

Panel to Change	How to Change It
Hardcopy Output Characteristics	Select option 0 (Hardcopy Output and JOB Statement Definition) from the ASG-JOB/SCAN Primary Menu.
Job Statement for Batch Submission	Select option 0 (Hardcopy Output and JOB Statements Definition) from the ASG-JOB/SCAN Primary Menu and press End.
Processing Options	Select Processing Options from the ASG-JOB/SCAN Primary Menu or type P on the command line of the ASG-JOB/SCAN - JCL Validation panel.

### Hardcopy Output Characteristics Panel

Designate options for TSO hardcopy on the Hardcopy Output Characteristics panel. When field entries for this panel are completed:

1. Press Enter to verify the entries. If any invalid entries are found, an error message appears in the upper right of the panel and the cursor moves to the field in error.
2. Correct the error, and press Enter again.
3. Repeat steps 1 and 2 until no more errors are found.
4. Press End to proceed to the Job Statement for Batch Submission panel.

### Panel Fields

Table explains what to enter in the required and optional fields of the Hardcopy Output Characteristics panel. In most cases, each value corresponds to an equivalent JCL parameter.

Hardcopy Output Characteristics panel fields

Field	Required/Optional	Description
FORCE OPE HARDCOPY FOR INITIALIZATION	Optional	Tells whether or not to print a hardcopy of the Options in Effect report each time JOB/SCAN is executed. The report shows the Run and Report Options in effect. Type Y for yes or N for no (the default is N).

Field	Required/Optional	Description
CLASS	Required	Designates the output class for hardcopy output. Because JOB/SCAN cannot verify the class before execution, make sure the class is valid for the site.
DEST	Optional	Specifies a destination value for hardcopy output. This field is compatible with the JES length and structure.
COPIES	Optional	Specifies the number of copies (from 1 to 255 for JES2, or 1 to 254 for JES3) for hardcopy output. The default is 1.
FCB	Optional	Specifies the name of the forms control buffer image JES is to use to guide printing of this SYSOUT dataset.
FORMS	Optional	Designates special forms to be used for hardcopy output requested during foreground processing.
UCS	Optional	Used for printers other than the IBM 3800. This parameter is mutually exclusive with the CHARS field.
CHARS	Optional	Used only if an IBM 3800 graphics font is installed. This parameter is mutually exclusive with the UCS field.
FORMDEF	Optional	<p>Identifies a library member that contains statements to tell the Print Services Facility (PSF) how to print the SYSOUT dataset on a page-mode printer. This relates to overlay forms, location of overlays, and suppressions that can be activated for certain page formats.</p> <p>If you normally enter a FORMDEF parameter on an OUTPUT statement in your JCL, enter the FORMDEF value here. Otherwise, leave this field blank.</p>
PAGEDEF	Optional	<p>Identifies a library member that contains statements to tell the Print Services Facility (PSF) how to print the SYSOUT dataset on a page-mode printer.</p> <p>If you normally enter a PAGEDEF parameter on an OUTPUT statement in your JCL, enter the PAGEDEF value here. Otherwise, leave this field blank.</p>
HOLD	Optional	Tells whether or not to hold SYSOUTs for TSO/E output processing. Type <b>Y</b> for yes or <b>N</b> for no.
HARDCOPY PANEL	Optional	Type <b>Y</b> for this panel to appear when Hardcopy Output is

Field	Required/Optional	Description
PROMPT		specified on a Foreground execution panel. Otherwise, leave the default of N.

**Hardcopy Output Characteristics panel fields**

```

----- HARDCOPY OUTPUT CHARACTERISTICS ----- ASG-JOB/SCAN
COMMAND  ===>

Force OPE hardcopy for Initialization ===> N (Y/N)

Enter/verify values for Hardcopy output:
Class    ===> A
Dest     ===>
Copies   ===> 1      1-255
FCB      ===>
Forms    ===>
UCS      ===>      (For other than IBM 3800)
Chars    ===>      (Only for IBM 3800)
FSSData  ===>
FormDef  ===>
PageDef  ===>

Hold     ===> Y      (Y/N)
    
```

**Job Statement for Batch Submission Panel**

Use the Job Statement for Batch Submission panel to specify:

- A JOB statement for batch execution.
- Step accounting information.
- Whether this panel should be displayed prior to every batch submission.

**To use the Job Statement for Batch Submission panel**

1. Complete the field entries.
2. Press Enter to verify the entries. If any invalid entries are found, an error message appears in the upper right of the panel and the cursor moves to the field in error.
3. Correct the error and press Enter again.
4. Repeat steps 2 and 3 until no more errors are found.
5. Press End to return to the ASG-JOB/SCAN Primary Menu.

**Completing the JOB Statements Section**

When filling in the JOB statement lines, follow these guidelines:

- All five lines of this section must be coded, even if some are comment lines.
- Lines may consist of a JOB statement, JES statements, or any other statements that are valid before the first EXEC statement in a job. These lines are inserted before an execute PROC statement, which is generated when a batch run is selected from the panels.

- The information must conform to valid JCL, JES2, or JES3 syntax rules.
- If a job name is not coded, it defaults to the TSO userid with the suffix “1” the first time JOB/SCAN is executed.
- Use site conventions for specifying the correct job name, accounting field, and other JCL parameters.

### Completing the Step Accounting Section

When filling in the ACCT field, follow these guidelines:

- Specify up to 61 characters for the ACCT parameter on the EXEC statement to be generated for batch execution.
- Leave this field blank if step accounting is not being used.

### Specifying Panel Prompt

*To be prompted with the Job Statement for Batch Submission panel prior to every execution of JOB/SCAN in panel-assisted batch mode*

- Type Y at the prompt. Otherwise, leave the default of N.

If JOB/SCAN is to be validated in batch mode, complete the Job statements section of the panel.

#### Job Statement for Batch Submission panel

Job statements section

```

----- JOB STATEMENT FOR BATCH SUBMISSION ----- (2 of 2) --
COMMAND   ===>

JOB STATEMENT INFORMATION: (Required - Enter/Verify JOB control statements)
===> //DSSCOM2 JOB (ACCOUNT),'USER NAME      ',
===> //      NOTIFY=DSSCOM2
===> //*
===> //*
===> //*

Step accounting information for generated EXEC statement (61 chars max)
ACCT   ===>

If you wish to be prompted with this panel when the Batch option
is specified, enter Y below:
JOB Statement panel prompt ===> N (Y/N)

Press <END> to return.

```

### Processing Options Panel

This panel allows you to tailor the Run Options for each user, as necessary. When JOB/SCAN is executed using ISPF panels, the values entered on this panel override any Run Options parameters coded in members in the SJJSPARM library.

**To use the Processing Options panel**

1. Complete the field entries.
2. Press Enter to verify entries. If any invalid entries are found, an error message appears in the upper right of the panel and the cursor moves to the field in error.
3. Correct the error and press Enter again.
4. Repeat steps 2 and 3 until no more errors are found.
5. Press End to proceed to the ASG-JOB/SCAN Primary Menu.

**Panel Fields**

Table explains what to enter in the fields of the Processing Options panel.

Processing Options panel fields

Field	Description
USE ERROR TABLE	Type <b>Y</b> to access Control Table AW for error suppression based on severity levels. The default is <b>N</b> .
CONDITION CODE PROCESSING	Type <b>Y</b> to request that JOB/SCAN evaluate all COND parameters and // IF statements and process the JCL accordingly. Entering <b>Y</b> also uses Control Tables AU and AR for step processing and return code simulation. The default is <b>N</b> .
CATALOG ACCESS	Usually, type <b>Y</b> for this parameter. Type <b>N</b> to inhibit access to the system catalog. The default is <b>Y</b> .
VTOC ACCESS	Usually, type <b>Y</b> for this parameter. Type <b>N</b> to inhibit access to the DASD VTOCs. The default is <b>Y</b> .
PDS MEMBER VERIFICATION	Type <b>Y</b> to verify the existence of PDS members referenced in the JCL. If <b>Y</b> is entered, the VTOC Access parameter must also be set to <b>Y</b> . Verification is <i>not</i> done for the following: <ul style="list-style-type: none"> <li>• Datasets with a disposition of <b>NEW</b></li> <li>• Control card datasets if the Access Control Cards parameter is <b>Y</b> and a JLJPF DD statement has been coded in the JOB/SCAN JCL, TSO logon PROC, or in a File Allocations Run Option statement. The default is <b>Y</b>.</li> </ul>
SECURITY EXIT DSNAME CHECK	Type <b>Y</b> if the security exit is to check each dataset name in the input JCL for possible security violations. If there is no security exit active or no DSNAME-checking support in the exit, this option is ignored.
PROGRAM CHECKING	Type <b>Y</b> to cause JOB/SCAN to validate the existence of programs. <b>N</b> turns off this validation. The default is <b>Y</b> . Coding <b>Y</b> for this parameter and <b>N</b> on either Catalog Access or VTOC Access, results in a volume mount request if the JOBLIB or STEPLIB

Field	Description
	volume is not on-line to the system.
DISABLE SMS PROCESSING	Type <code>Y</code> to bypass all SMS processing. If <code>N</code> is specified, the options specified in the SMS Run Option are to be in effect for this JOB/SCAN run. This variable is ignored if the SMS subsystem is not active during JOB/SCAN. The default is <code>N</code> .
USE SCHEDULER INTERFACE	Type <code>Y</code> to turn on the scheduler interface specified by the SCHEDULER run option. Type <code>N</code> to disable scheduler interfaces. The use of this option is fully explained in <a href="#">User Information - JOB/SCAN with Schedulers</a> .
IDCAMS PROCESSING	Type <code>Y</code> to cause validation to be performed on IDCAMS control statements. <code>N</code> turns off this validation. The Catalog Access and VTOC Access parameters must be set to <code>Y</code> if IDCAMS statements are not in-stream. The default is <code>Y</code> .
JOB LIMIT	Enter the maximum number of jobs to be processed (1-999) in a single JOB/SCAN execution. The default is "no limit".
ACCESS CONTROL CARDS	Type <code>Y</code> to access control card libraries, verify control card input, and include control cards in the Structured JCL Listing (SJL). The Catalog Access and VTOC Access parameters must be set to <code>Y</code> if this parameter is set to <code>Y</code> . If this parameter is set to <code>Y</code> , any dataset names in Control Table AQ or ddnames in Control Table AB are treated as control card input. Any dataset names in Control Table CG will not be expanded in the SJL. The default is <code>Y</code> .
CONTROL CARD LIMIT	Limits the number of control cards (1-999) listed in the SJL for each ddname or dataset name identified as control card input. The default is blank, which means "no limit". This parameter is ignored if Access Control Cards is set to <code>N</code> .
SUBSYSTEM	Use <code>JES2</code> , <code>JES3</code> , or <code>NONE</code> to indicate whether the JCL contains JES2 or JES3 statements to be validated for syntax. <code>NONE</code> tells JOB/SCAN not to evaluate JES statements. The default is <code>JES2</code> .
SUPPRESS ERROR MESSAGES	Usually, type <code>NO</code> to include all messages in the SJL. Type <code>A</code> to suppress advisory messages; <code>W</code> to suppress warning and advisory messages; or <code>E</code> to suppress advisory, warning, and error messages. Alternatively, use a number from 0 to 99 to suppress all error messages with a severity level equal to or less than the number coded. This option is independent of Control Table AW, which can suppress specific errors. The default is <code>NO</code> .
EXTENDED ERROR TEXT	Type <code>X</code> to cause the extended error message text to be printed for each error listed on the SJL. The default is <code>S</code> , causing only the short error message to be included.

The Processing Options panel is used to override specific Run and Report Options.

## Processing Options panel

```

D02PEP ----- JOB/SCAN PROCESSING OPTIONS -----
COMMAND ==>

Use Error Table          ==> N          (Y/N)
Condition Code Processing ==> N          (Y/N)
Catalog Access          ==> Y          (Y/N)
VTOC Access             ==> Y          (Y/N)
PDS Member Verification ==> Y          (Y/N)
Security Exit DSNAMES Check ==> Y      (Y/N)
Program Checking         ==> Y          (Y/N)
Disable SMS Processing   ==> N          (Y/N)
IDCAMS Processing       ==> Y          (Y/N)
Use SCHEDULER interface ==> N          (Y/N)
Job Limit               ==>           (1 to 999 or Blank For No Limit)
Access Control Cards     ==> Y          (Y/N)
Control Card Limit      ==>           (1 to 999 or Blank For No Limit)
Subsystem               ==> JES2      (JES2/JES3/None)
Suppress Error Messages ==> NO        (A,W,E,0 to 99,or NO)
Extended Error Text     ==> S          (X=Extended,S=Short)

```

# Running the Product in an APF-authorized Environment

This topic provides guidelines for running in an APF-authorized environment.

- [When is an APF-authorized Environment Required?](#)
- [Setting Up the ISPF Environment](#)
- [Setting Up Batch Processing](#)
- [Security Exit Considerations](#)

## When is an APF-authorized Environment Required?

JOB/SCAN and DOCU/TEXT require an APF-authorized environment for these optional functions:

- Verifying the SCHENV JOB statement parameter. The product invokes the Workload Management Scheduling Environments Determine Execution Service to verify that the scheduling environment exists and is available to the current system.
- Querying a volume within the tape storage environment. For example, when a DD statement refers to an uncataloged dataset and the VOL=SER parameter is specified but the UNIT parameter is omitted, the product will query the Library Control System (LCS) to see if this volume is in a tape library.
- Verifying access to datasets when the USERID parameter from the JCL being processed is different from the USERID of the product job or TSO session. Refer to the information on security exit interfaces in [Setting Up Optional Interfaces and Exits](#) for more information.



The use of the USERID parameter from the JCL is only available when the JOB/SCAN or DOCU/TEXT program being executed has access to the library that it will process. If the product does not have access to a library, the library will not be opened even if the USERID on the job card has access to the library.

- Running SMS.
  - If the DFSMS subsystem uses the USERID in the ACS routines to determine SMS constructs, and the USERID in the JCL being scanned is different than the USERID of the JOB/SCAN or DOCU/TEXT job or TSO session.
  - For non-APF, the product ignores the USERID of the JCL being scanned, and the USERID of the JOB/SCAN or DOCU/TEXT job or TSO session is used, possibly causing unpredictable results.

## Setting Up the ISPF Environment

If TSO/E Release 2 or later is installed, the product can perform the APF-authorized functions under ISPF.



**To run APF-authorized in an ISPF environment**

- Add the name `J00YAAI` to the `SYS1.PARMLIB` member `IKJTSOxx` under sections `AUTHPGM` and `AUTHTSF`. You might need a systems programmer to perform this function.



The default setting of the ISPF SPECIFICATIONS Run Option is `AUTHFUNCTIONS`, allowing the product to invoke APF-authorized functions. To bypass APF-authorized functions and improve system performance, set the Run Option to `NOAUTHFUNCTIONS`.

## Setting Up Batch Processing

`JOB/SCAN` and `DOCU/TEXT` run in an APF-authorized environment by following these MVS rules:

- All load modules must reside in APF-authorized libraries.
- The `JOBLIB/STEPLIB` DD concatenation can specify *only* APF-authorized libraries in the product JCL.
- The product checks the datasets allocated to the ddname `JLLOD` for exits such as the Selection Exit, Report Exit, Standards Exit, and Security Exit. The `JLLOD` DD concatenation must specify *only* APF-authorized libraries in the product JCL.

## Security Exit Considerations

If you are using the product Run Option called `SECURITY EXIT=xxxxx,DSNAME` and are checking JCL that executes under a `USERID` that is not the `USERID` of the `JOB/SCAN` or `DOCU/TEXT` job or current TSO session, the Security Exit named `xxxxx` must be in an APF-authorized environment.

### Security Exit Module

ASG supplies Security Exit module `SAF2`. It is shipped as source code in the `SJSSAMP` library, as well as a load module in the `SJSLMOD` library.

`SAF2` is written to use the `REQUEST=AUTH` and `STATUS=ACCESS` parameters of the `RACROUTE` macro to interface with the System Authorization Facility (SAF). `SAF2` does not log security violations when the user does not have proper access authority.

Because under ISPF subpool 0 storage is deleted after the module is executed, `SAF2` uses subpool 251 for its storage if it is invoked APF-Authorized. Subpool 251 storage is not deleted until the job step task terminates. This is a restriction of ISPF's APF-authorization processing.

`CA-ACF2` requires APF-authorization to use `STATUS=ACCESS` unless a (CA-ACF2) `SAFDEF` record is defined with the `NOAPFCHK` keyword.

## Enabling DOCU/TEXT

DOCU/TEXT is a password-enabled component of JOB/SCAN that provides additional JCL-management features. With DOCU/TEXT, you can generate reports that document systems and jobs, create job-related User Text, and perform cross-reference queries that list where-used information about PROCs, programs, files, and SYSOUTs.



You can run DOCU/TEXT in conjunction with JOB/SCAN, or as a stand-alone product (if licensed).

This topic provides the requirements and instructions for setting up and customizing DOCU/TEXT. DOCU/TEXT and JOB/SCAN share many of the same product libraries. As a result, the JOB/SCAN installation also initializes and allocates many of the files required to run DOCU/TEXT. However, some additional set up and configuration is required to enable DOCU/TEXT-specific functions.

If you are licensed for DOCU/TEXT, identify which components you plan to implement. These modules are available:

- ISPF Primary Menu Option
  - ONLINE INFO
- ONLINE XREF
- DOCU/TEXT Analysis Documents
- DOCU/TEXT Operations Documents

The instructions in this topic explain how to customize DOCU/TEXT. As conditions and requirements change at your site, you can change the initial settings.

- [Installing the TSO Online Facility](#)
- [Adding DOCU/TEXT to the ISPF Primary Menu](#)
- [Allocating the Files](#)
- [Applying Product Authorization Code\(s\)](#)
- [Running DOCU/TEXT in an APF-authorized Environment](#)
- [Creating the DOCU/TEXT Environment](#)
- [Allocating PDS Storage](#)
- [Setting Up Optional Interfaces and Exits](#)
- [Final Installation Tasks](#)

## Installing the TSO Online Facility

There are three ways to make the DOCU/TEXT libraries available from the ISPF Primary Option Menu:

- **Method 1.** Modify the existing TSO logon PROC. Use this method unless the logon PROC is not easily accessible.

- **Method 2.** Modify ALLOCDT in the SJJSCSLST library. Use this method as a temporary measure until a new logon PROC can be created.
- **Method 3.** Modify LIBDEFDT in the SJJSCSLST library. Use this method if the DOCU/TEXT libraries are allocated using the ISPF LIBDEF service.

### Method 1—Modify the TSO Logon PROC

#### To modify the existing TSO logon PROC

1. Concatenate (ahead of any other datasets) the DOCU/TEXT datasets for these ddnames:
  - SYSPROC (CLISTs)
  - ISPLLIB (load library—may be replaced with a STEPLIB)
  - ISPPPLIB (ISPF panels)
  - ISPMLIB (ISPF messages)

If the LE runtime libraries CEE.SCEERUN and CEE.SCEERUN2 are not in your LINKLIST, add them to ISPPLIB (or STEPLIB).

If the BLKSIZE parameter is not coded for each ddname, ensure that the first dataset in the concatenation has a block size equal to or greater than the largest dataset in the group. Code a BLKSIZE parameter for the first dataset.



Ensure that no member names are duplicated among the DOCU/TEXT datasets, IBM-provided datasets, or site-specific datasets.

2. Add DD name JLRCL. JLRCL should be allocated to the SJJSPARM library. For example:

```
//JLRCL DD DISP=SHR.DSN=X.X.SJJSPARM
```

3. Because the TSO logon PROC has been changed, log off the terminal session and log on again using the modified logon PROC.

### Method 2—Modify ALLOCDT CLIST



If you choose this method, you must execute the CLIST before entering ISPF each time you run DOCU/TEXT.

#### To modify member ALLOCDT in the SJJSCSLST library

1. In the PROC statement, change the Xs in parentheses after DJSNAME to the high-level qualifier used when loading the DOCU/TEXT libraries. For example:

```
PROC 0 DJSNAME(ASG.JSnnn)
```

where *nnn* is the three-character DOCU/TEXT version.

2. Add any other site-required ISPF library concatenations to the appropriate TSO ALLOC commands. (Refer to current logon PROCs.). If the LE runtime libraries CEE.SCEERUN and CEE.SCEERUN2 are not in your LINKLIST, add them to ISPPLIB (or STEPLIB).

3. Verify that the DOCU/TEXT datasets have the same or greater block size as other ISPF datasets in the concatenated sequences.
4. Because you modified the ALLOCDT member in the SJJSCSLST library, return to TSO READY mode and enter this command to execute the CLIST:

```
EXEC 'ASG.JSnnn.SJJSCSLST(ALLOCDT)'
```

where *ASG.JSnnn* is an example of a site-specific high-level qualifier, and *nnn* is the three-character DOCU/TEXT version.

### Method 3—Modify LIBDEFDT CLIST

If your site allocates the DOCU/TEXT libraries using the ISPF LIBDEF service, modify the LIBDEFDT member in the SJJSCSLST library.

- Change the value of DSJNAME to the appropriate DOCU/TEXT dataset prefix. For example:

```
DSJNAME = "ASG.JSnnn"
```

where *ASG.JSnnn* is an example of a site-specific high-level qualifier, and *nnn* is the three-character DOCU/TEXT version.

The LIBDEFDT member in the SJJSCSLST library is automatically executed each time that you select the DOCU/TEXT option from the ISPF Primary Option Menu.



The ISPF LIBDEF allocation service will temporarily add the DOCU/TEXT libraries to the current concatenations. LIBDEFDT will remove the DOCU/TEXT libraries from the concatenations when you exit DOCU/TEXT.

### Modifying the LIBDEFDT member in the SJJSCSLST library

```
/* rexx */
DJSNAME = "ASG.JSnnn"
Parmlib = "'DJSNAME'.SJJSPARM'"
address TSO

/* Allocate JLLOD */
rc = LISTDSI("JLLOD" "FILE")
if rc <> 0 then do
    "alloc f(JLLOD) da('DjsName'.SJJSLMOD) reu shr" /* is JLLOD allocated? */
    lodalc = "Y"
end
```

## Adding DOCU/TEXT to the ISPF Primary Menu

You can add DOCU/TEXT as an option on the ISPF Primary Menu.

**To add DOCU/TEXT to the ISPF Primary Option Menu**

1. Locate the panel called ISR@PRIM in the existing IBM-supplied ISPLIB library.
2. Copy ISR@PRIM to the JOB/SCAN SJSPLIB library.
3. Add DOCU/TEXT to the existing list of menu items in the ISR@PRIM panel:

```
% D +DOCU/TEXT ISPF Support Environment
```

4. Change the menu command definition to add the product to the ISPF Primary Option Menu.

For LIBDEF allocation, insert this line:

```
D, 'CMD (LIBDEFDT) '
```

For non-LIBDEF allocation, insert this line:

```
D, 'PGM (D00YDMU) NEWAPPL (DT)'
```

**LIBDEF Example**

In this example, the DOCU/TEXT ISPF support environment indicated by `D, 'CMD (LIBDEFDT) '` has been added to the ISPF Primary Option Menu.

**Change ISPF Primary Option Menu—LIBDEF method**

```
1, 'PGM (ISRBRO)      PARM (ISRBR001) '
2, 'PGM (ISREDIT)    PARM (P, ISREDM01) '
3, 'PANEL (ISRUTIL) '
4, 'PANEL (ISRFPA) '
5, 'PGM (ISRJB1)     PARM (ISRJPA) NOCHECK '
6, 'PGM (ISRPTC) '
7, 'PGM (ISRYXDR)    PARM (ISR)      NOCHECK '
8, 'PANEL (ISRLPRIM) '
9, 'PANEL (ZSDSFOP2) NEWAPPL (ISF) '
D, 'CMD (LIBDEFDT) '
T, 'PGM (ISPTUTOR)   PARM (ISR00000) '
', ' '
X, 'EXIT'
```

**Non-LIBDEF Example**

In this example, the DOCU/TEXT ISPF support environment indicated by `D, 'PGM (D00YDMU) NEWAPPL (DT) '` has been added to the ISPF Primary Option Menu.

**Change ISPF Primary Option Menu—non-LIBDEF method**

```
1, 'PGM (ISRBRO)      PARM (ISRBR001) '
2, 'PGM (ISREDIT)    PARM (P, ISREDM01) '
3, 'PANEL (ISRUTIL) '
4, 'PANEL (ISRFPA) '
5, 'PGM (ISRJB1)     PARM (ISRJPA) NOCHECK '
6, 'PGM (ISRPTC) '
7, 'PGM (ISRYXDR)    PARM (ISR)      NOCHECK '
8, 'PANEL (ISRLPRIM) '
9, 'PANEL (ZSDSFOP2) NEWAPPL (ISF) '
D, 'PGM (D00YDMU)    NEWAPPL (DSS) '
T, 'PGM (ISPTUTOR)   PARM (ISR00000) '
', ' '
X, 'EXIT'
```

## Adding ONLINE INFO to the ISPF Primary Menu

ONLINE INFO is available from the DOCU/TEXT primary menu. However, you have the option to access ONLINE INFO directly from your ISPF Primary Option Menu.

### To add ONLINE INFO to the ISPF Primary Option Menu

1. Change the menu command definition to add ONLINE INFO to the ISPF Primary Option Menu:

```
L, 'PGM(D00YDMU) PARM(OLIQ) NEWAPPL(DSS)'
```

2. If desired, specify a jobname within the command definition:

```
L, 'PGM(D00YDMU) PARM(OLIQ,jobname) NEWAPPL(DSS)'
```

## Allocating the Files

Perform file allocations based on the method you used to install the TSO online facility.

- If you changed the TSO logon PROC, log off from the terminal session and log on again using the modified logon PROC.
- If you modified ALLOCDT in the SJJSCST library, return to TSO READY mode and enter this command to execute the CLIST:

```
EXEC 'X.X.SJJSCST(ALLOCDT)'
```

where *X.X* is the qualifier associated with your library.



For this method, you must execute the CLIST before entering ISPF each time you use DOCU/TEXT.

- If you modified LIBDEFDT in the SJJSCST library and specified CMD(LIBDEFDT) in the ISPF Primary Option Menu panel definition, the libraries are allocated each time you access DOCU/TEXT.

## Applying Product Authorization Code(s)

In order to activate DOCU/TEXT, you must install your product authorization code(s) into the appropriate load module. See [Applying Product Authorization Code\(s\)](#) for instructions on how to assemble and link these codes.

## Running DOCU/TEXT in an APF-authorized Environment

The DOCU/TEXT functions that require APF-authorization are the same as those required for JOB/SCAN. Refer to [Running the Product in an APF-authorized Environment](#) for the functions that require DOCU/TEXT to run APF-authorized.

### Setting Up the ISPF Environment

If TSO/E Release 2 or later is installed, DOCU/TEXT can perform the APF-authorized functions under ISPF.

**To run APF-authorized in an ISPF environment**

- Add the name `D00YAAI` to the `SYS1.PARMLIB` member `IKJTSOxx` under sections `AUTHPGM` and `AUTHTSF`. You might need a systems programmer to perform this function.



The default setting of the ISPF SPECIFICATIONS Run Option is `AUTHFUNCTIONS`, allowing DOCU/TEXT to invoke APF-authorized functions. To bypass APF-authorized functions and improve system performance, set the Run Option to `NOAUTHFUNCTIONS`.

## Creating the DOCU/TEXT Environment

Setting up the DOCU/TEXT environment involves these tasks:

### Step 1—Modify Detail Statements in `DOCOPT01`

Modify detail statements to specify the `TITLE1` value and the user ID of the product administrator. The `DOCOPT01` member in the `SJJSPARM` library contains control information. Update the `TITLE1` value and set up access authorization for the product administrator.

### Changing the `TITLE1` Value

The `TITLE1` value is usually the your company name. This value appears on printed report headings.

#### To change the `TITLE1` value

- Locate the System Identification group-level statement in `DOCOPT01` in the `SJJSPARM` library.
- Edit the alphanumeric field to the right of `TITLE1`. Enclose the name in single quotes. For example:

```
SYSTEM IDENTIFICATION
    TITLE1='up to 32 characters'
```

### Assigning Access to the Product Administrator

The Access Control statements specify access to online panels. This is where the product administrator's user ID is specified.



If DOCU/TEXT will not be invoked under ISPF, delete or comment out the Access Control group-level statement and all Access Control detail-level statements.

#### To assign Product Administrator access

- Locate the Access Control group-level statement in `DOCOPT01` in the `SJJSPARM` library.
- Edit the `ADMINISTRATOR=USERID` detail-level statement to reflect the user ID of the product administrator for your site. For example:

```
ACCESS CONTROL
    ADMINISTRATOR=USERID=myid
```

## Step 2—Specify PROCLIBs in DOCOPT02

The DOCOPT02 member in the SJJSPARM library contains the File Allocations statements that enable DOCU/TEXT to find the necessary datasets. This is where the site-specific PROCLIBs are specified. PROCLIBs are searched in the order that libraries are specified.

### To modify File Allocations

- a. Locate the File Allocations group-level statement. For example:

```
FILE ALLOCATIONS
JLJPA  =SHR, DSN=SYS1 . PROCLIB
      =SHR, DSN=SYS2 . PROCLIB
      =SHR, DSN=SYS3 . PROCLIB, UNIT=SYSDA, VOLSER=TEST01
      =SHR, DSN=SYS3 . PROCLIB, UNIT=SYSDA, VOLSER=PROD01
```

- b. Replace the dataset names with the site-specific PROCLIB names. The UNIT and VOLSER parameters are required only for uncataloged datasets.



Because the JLJPA PROCLIB TYPE defaults to PDS, if the PROCLIB type is PANVALET, LIBRARIAN, or LIBRARIAN-3, then code the JLJPA PROCLIB TYPE Run Option statement. See Run Option in ASG-DOCU/TEXT's System Administration guide for more information.

- c. Add or delete PROCLIBs as required.
- d. If the BLKSIZE parameter is not coded, ensure that the dataset with the largest block size is first in the concatenation. Alternatively, code the BLKSIZE parameter on the first dataset in the concatenation, specifying the largest block size as its value. For example:

```
JLJPA=SHR, DSN=TEST . PROCLIB, BLKSIZE=13200
```

Use the same concatenation sequence as the site JES2 or JES3 concatenation. See Use of File Allocation Statements in ASG-DOCU/TEXT's System Administration guide for a list of valid parameters for the File Allocation statements. For information about processing JES2 and JES3 control statements, see [Rules for JES2 and JES3 Control Statements](#).

## Step 3—Modify the Run and Report Options in DOCOPT02

The DOCOPT02 member in the SJJSPARM library contains Run Options, Report Options, and Control Tables (discussed in [Step 4—Modify Control Tables in DOCOPT02](#)).

- **Run Options.** These statements control the way DOCU/TEXT processes the JCL and how information is displayed in reports.
- **Report Options.** These statements control the content and layout of DOCU/TEXT reports, as well as whether each report is to be printed or displayed.

See Run Options and Report Options in ASG-DOCU/TEXT's System Administration guide for more information.



**To review Run Options and Report Options**

- Locate the Run Options and Report Options group-level statements in DOCOPT02, and review the detail statements to determine whether the entries are appropriate for your site.

**To change the format in which dataset names are specified on panels**

- Specify one of these Run Options:

```
ISPF SPECIFICATIONS=DSNFORMAT=TSO
```

**Or**

```
ISPF SPECIFICATIONS=DSNFORMAT=NONTSO
```

The default value is TSO (TSO conventions). Under this setup, DOCU/TEXT attaches the user prefix as the first node of any dataset name not enclosed in quotes.



When changing DSNFORMAT to NONTSO, always enter the fully qualified dataset name on the panels and do not use quotes.

**Step 4—Modify Control Tables in DOCOPT02**

Control Tables reflect the site environment. At this point in the installation process, you might want to change values in these tables:

- **Control Table AA.** This table associates UNIT names with device codes, which indicate the types of devices. Alter entries to override device codes and define unit values as appropriate.
- **Control Table AQ.** This table specifies sort and utility control card library names. Add to or change existing entries to control the content of DOCU/TEXT reports as appropriate. If SYSIN is the ddname for these libraries, Control Table AQ changes are not required.

For a description of the run options associated with Control Table AA and Control Table AQ, see [Run Options for Control Tables AA and AQ](#).

For a description of the values associated with Control Table AA, see [Control Table AA Values](#).

See Control Tables in DOCU/TEXT in ASG-DOCU/TEXT's System Administration guide for complete information about Control Tables.

**Modifying Control Table AA Values**

If IBM's MVS module IEFEB4UV is available and the Run Options Catalog Access Control (CAC=YES,VTOC) is specified, DOCU/TEXT uses the Eligible Device Table (EDT) to validate the UNIT parameter in the JCL and to select the device code. In this case, no alterations to Control Table AA are necessary.

You must alter entries in Control Table AA when performing these functions:

- Overriding the device code for a given unit, such as 3380.
- Defining valid unit values and device codes if CAC=NO or module IEFEB4UV is not available.

**To change Control Table AA values**

- a. Locate the Control Tables group-level statement in the DOCOPT02 member in the SJJSPARM library.

The entries beginning with AA associate unit names with device codes. Use the existing entries as a pattern: The two As are placed in columns 6 and 7; the device type, name, or unit address begins in column 8; and the device code is placed in column 21. For example:

```
-----+-----1-----+-----2-----+-----3-----+-----4
AACART                                K
```

- b. Delete Control Table AA entries that are not needed. Add other names used at your site.



If you delete entries (example, AADISK) and do not code unit names in the IDCAMS control statement, you will need to code the Default Unit Names (DUN) Run Option. See Default Unit Names (DUN) in ASG-DOCU/TEXT's System Administration guide for additional information about this requirement.

### Modifying Control Table AQ Values

The Control Table AQ values determine which libraries contain control statements and should print in appropriate DOCU/TEXT reports.

Add or change existing entries in this format:

```
-----+-----1-----+-----2-----+-----3-----+-----4
CONTROL TABLES
AQSYSQ.PARMLIB
```

#### Step 5—Modify the PROC that runs DOCU/TEXT in batch mode

If DOCU/TEXT will be run in batch, you must modify the DOCU/TEXT PROC (DT5PROC) in the SJJSSAMP library to specify the appropriate load library and SJJSPARM library members.

If DOCU/TEXT queries will be run in batch, modify PROC DT5XBQMF in SAMPLIB.



If you plan to execute DOCU/TEXT or DOCU/TEXT queries as a submitted batch job, modify the JOB statement in the DTMVS member in the SJJSSAMP library as necessary.

#### To insert high-level qualifiers for the LOAD library

- a. Locate the DT5PROC member in the SJJSSAMP library.
- b. Change the value in the DJSNAME with the first two nodes of the dataset names containing your DOCU/TEXT libraries (that is, the high-level qualifiers used when LOADALL was submitted). For example:

```
//DT5PROC PROC DJSNAME='xxxx.xxxx'
```

If you followed standard conventions, node 1 is unique to your site and node 2 is equal to the TRLSE value.

#### To change SJJSPARM members

- a. Locate the OPT01, OPT02, and OPT0N symbolics in DT5PROC. For example:

```
//DT5PROC PROC DJSNAME='xxxx.xxxx'.
OPT01=DOCOPT01.
```

```
OPT02=DOCOPT02.
OPT0N=DOCOPT03
```

- b. When executing DOCU/TEXT, you can change OPT0N to point to any DOCOPT member. Use this table for guidance when changing DOCOPT members:

If You Are Installing:	Specify this OPT0N Statement:
Both Analysis Documents and Operations Documents	OPT0N=DOCOPT05
Only Analysis Documents	Make no changes
Only Operations Documents or only ONLINE INFO	OPT0N=DOCOPT04

- c. Store the PROCs in a PROCLIB accessible to all users.

### To run DOCU/TEXT queries in batch

- Modify the PROC from member DT5XBQMF in the SJSSAMP library with the location of the ASG LOAD library and the OLX repository.

### Modifying DT5XBQMF

```

//*****
//*   OLX BATCH QUERY PROC - EXECUTED BY OLX QUERY ISPF PANEL.
//*
//*   REF:   SAMPLIB (DT5XBQMF)
//*
//*   NOTE:  MODIFY ITEMS INDICATED BY ARROWS <==
//*****
//IFXBQMF  PROC
//STEP01   EXEC  PGM=D50XBQX
//STEPLIB  DD   DISP=SHR,DSN=DSSI.LOAD           <== YOUR ASG LOAD LIB
//SYSPRINT DD   SYSOUT=*,BLKSIZE=132,LRECL=132
//SYSOUT   DD   SYSOUT=*
//JLJDOC   DD   SYSOUT=*
//JLLTM    DD   DISP=SHR,DSN=DSSI.OLX.FILE      <== YOUR OLX REPOSITORY

```

### Step 6—Install the edit macros for the DOCU/TEXT User Text facility

The D40CUTEM EDIT macro is provided in the SJJSCLST library. It is utilized by the DOCU/TEXT User Text facility, and is called internally by the product (not entered on the command line).

#### To install the edit macro

- Ensure that the product SJJSCLST library is allocated in the SYSPROC concatenation.
- Make no changes to D40CUTEM.

### Step 7—Create a VSAM file to use Short Descriptions and ONLINE XREF

If you plan to use the Short Descriptions facility (SDF) and/or the ONLINE XREF (OLX) facility to perform online cross-reference queries, you must create a VSAM SDF/OLX file (also referred to as a *repository*) that contains *where-used* information about jobs, PROCs, programs, files, and SYSOUTs. Cross-reference data and user-supplied Short Descriptions are stored in this VSAM file.

You can use this algorithm to estimate the storage required for ONLINE XREF. Replace the "Assumptions" with actual data at your site.

<b>Assumptions</b>	
Number of jobs added	6,536
Number of PROCs added	4,302
Number of programs added	3,327
Number of files added	45,754
Number of SYSOUTs added	4
<b>Space Calculations</b>	
Jobs A1	6,536
Average PROCs/Job A2	0.65820
Average Programs/Job A3	0.50903
Average Files/Job A4	7.00031
Average SYSOUTs/Job A5	0.00061
3380 bytes/track	47,476
3390 bytes/track	56,664

<b>Estimated Bytes</b>	$A1 \times (1 + A2 + A3 + A4 + A5) \times 300 = 17,976,900$
<b>Estimated Tracks</b>	379 "3380 tracks" = 47,476 bytes per track *
	317 "3390 tracks" = 56,664 bytes per track *

\* DASD space information was obtained from the *IBM 3390 Direct Access Storage Reference Summary, GX26-4577-2*.

For more information about allocating, customizing, loading, and maintaining the OLX repository, refer to ONLINE XREF: Creating and Maintaining the OLX Repository in ASG-DOCU/TEXT's User's guide.

**To create and edit the SDF/OLX file**

- a. Determine these items for the VSAM file:
  - A valid unit and volume serial number.
  - A dataset name.
  - The required size (calculated by the algorithm).
- b. Edit the DTVSAM member in the SJJSSAMP library.
- c. Modify the JOB statement to comply with your site's requirements.
- d. Examine each line indicated by an arrow (<==) and make any necessary changes.

**SAMPLIB member DTVSAM**

```
//DTVSAM JOB (ACCT), 'PROGRAMMER NAME'
//*****
//* CREATE THE SDF/OLX VSAM FILE TO BE USED FOR SHORT DESCRIPTIONS
//* AND/OR ONLINE XREF (OLX) INFORMATION
//* REF: SAMPLIB (DTVSAM)
//*
//* NOTE: MODIFY JOB STATEMENT AND ITEMS INDICATED BY ARROWS <==
//*****
//STEP01 EXEC PGM=IDCAMS,REGION=1024K
//SYSPRINT DD SYSOUT=*
//SDFOLX DD DISP=OLD,UNIT=SYSDA,VOL=SER=XXXXXX <== VALID UNIT/VOL
//SYSIN DD *
DEFINE CLUSTER -
    (NAME (YYYYYY.OLX.FILE) /* <== CLUSTER NAME */ -
    VOLUME (XXXXXX) /* <== VALID VOL */ -
    RECORDSIZE (500 22520) -
    INDEXED -
    KEYS (48 1) -
    SHAREOPTIONS (2) -
    FILE (SDFOLX)) -
DATA -
    (NAME (YYYYYY.OLX.FILE.DATA) /* <== DATA NAME */ -
    CYLINDERS (1000 100) /* <== DESIRED SIZE */ -
    CISZ (27648) /* <== CI SIZE */ -
    NOWRITECHECK) -
INDEX -
    (NAME (YYYYYY.OLX.FILE.INDEX)) /* <== INDEX NAME */
//
```

- e. Submit the JCL to create the file.

**To allocate the SDF/OLX file**

- a. Locate the JLLTM File Allocations statement in the DOCOPT02 member in the SJJSPARM library.
- b. Uncomment the statement by removing the asterisk in column 1.
- c. Replace the dataset name with the SDF/OLX dataset name.
- d. If you are in DOCU/TEXT, reinitialize to enable the allocation.

## Allocating PDS Storage

DOCU/TEXT uses two types of PDS libraries:

- PDSOUTs store DOCU/TEXT reports and other documentation for online display. Even if documents will not be displayed online, you must always allocate the PDSOUT named JLJPWOE so that the product administrator has access to the Test an Options Member function.
- User Text libraries are used to store optional user-written information that can be automatically included in the DOCU/TEXT reports and ONLINE INFO.

### PDSOUT Space Requirements

The PDSOUTS member in the SJJINST library contains the JCL for allocating PDSOUT and User Text libraries. The permanent space allocation suggestions in the PSPACE parameters are based on these assumptions:

- Number of jobs being documented per execution: 500
- Average job contents:
  - 1 procedure
  - 10 steps per procedure
  - 10 DD statements per step
  - 6 uses of control cards; 7 cards per member
  - 3 tape files; 1 passed to another step
  - 2 short steps
- Each job contains approximately 20 `/**` comment statements

If more than 500 jobs are documented during one execution, you might need to allocate larger PDSOUT libraries. If your space requirements differ from the above assumptions, make the necessary changes.

### The PDSOUT and User Text Files

The PDSOUT and User Text files to be allocated are listed in this section. The required files depend on the products being installed.

#### *To allocate the files*

- Submit the PDSOUTS member in the SJJINST library.

Guidelines for modifying the JCL to create or delete files are provided in [Editing the PDSOUTS Member](#).

This table will help you determine which DD statements you can comment out in the PDSOUTS allocation JCL. In cases where the same library is required for more than one product, only one allocation of that library is necessary.

Product Name	Dataset Name	Document Name	ddname
Analysis Documents	node1.PDSOUT.PWOE	Options in Effect	JLJPWOE
	node1.PDSOUT.PWTC	Table of Contents	JLJPWTC
	node1.PDSOUT.PWJL	Structured JCL Listing	JLJPWJL
	node1.PDSOUT.PWDX	Data Set XREF	JLJPWDX
	node1.PDSOUT.PWRX	Report XREF	JLJPWRX
	node1.PDSOUT.PWGX	Program XREF	JLJPWGX

Product Name	Dataset Name	Document Name	ddname
Operations Documents	node1.PDSOUT.PWCX	PROC XREF	JLJPWCX
	node1.PDSOUT.PWIL	I/O Listing	JLJPWIL
	node1.PDSOUT.PWFC	Flowcharts	JLJPWFC
	node1.PDSOUT.PWIX	Index	JLJPWIX
	node1.USERTEXT.UTGP	General Purpose User Text	JLJUD
	node1.USERTEXT.UTSK	User Text Skeletons	JLJUDDSK
	node1.PDSOUT.PWOE	Options in Effect	JLJPWOE
	node1.PDSOUT.PWJL	Structured JCL Listing	JLJPWJL
	node1.PDSOUT.PWJD	Job Documentation	JLJPWJD
	node1.PDSOUT.PWTU	Tape Usage Analysis	JLJPWTU
	node1.PDSOUT.PWTP	Tape Pull List	JLJPWTP
	node1.PDSOUT.PWTX	Tape VOLSER XREF	JLJPWTX
	node1.PDSOUT.PWRA	Report Analysis	JLJPWRA JLJUD
	node1.USERTEXT.UTGP	General Purpose User Text	JLJUDJD
node1.USERTEXT.UTJD	Job Documentation User Text	JLJUDDSK	
node1.USERTEXT.UTSK	User Text Skeletons	JLJUDDSK	
ONLINE INFO	node1.PDSOUT.PWOE	Options in Effect	JLJPWOE
	node1.PDSOUT.PWJD	Job Documentation	JLJPWJD
	node1.USERTEXT.UTJD	Job Documentation User Text	JLJUDJD
	node1.USERTEXT.UTSK	User Text Skeletons	JLJUDDSK
ONLINE XREF	node1.PDSOUT.PWOE	Options in Effect	JLJPWOE

### Editing the PDSOUTS Member

To allocate PDS storage, edit the PDSOUTS member in the SJJINST library using the guidelines provided in this section, and then submit the job. Review the output to ensure that the job ran successfully.



The same JCL can be modified to delete these libraries.

### Allocating the PDS for ONLINE XREF

Because ONLINE XREF users need to allocate only one work PDS, you can use these characteristics to allocate it:

- DSN: node1.PDSOUT.PWOE
- SPACE: (TRK,(1,2,1))
- RECFM=FB
- LRECL=132
- BLKSIZE=13200



The JCL in member PDSOUTS does not need to be submitted.

## Creating PDSOUT Files

### To create PDSOUT files

- Edit the PDSOUTS member according to this table:

Name	Description
Job statement	Copy the JOBCARD member from the SJJSINST library.
NODE1=xxxxxxxxx	Use the same high-level qualifier established earlier as the first node of the DJSNAME symbolic in the LOADALL JCL
NODE2=xxxxxx	Do not alter this value. It is PDSOUT for PDSOUT libraries and USERTEXT for User Text libraries.
DUNIT and DVOL	Specify a valid DASD unit and volume
ESPACE	This symbolic is set to a space allocation default sufficient for evaluating the DOCU/TEXT product. If you are installing the library for production use:  Comment out this symbolic: <code>ESPACE='10.5.5) ) '</code>  Uncomment out this symbolic: <code>ESPACE=,</code>
DDISP	Make sure that DDISP=(null) is the uncommented symbolic. This parameter is set to a NEW,CATLG disposition.
DD statements	Comment out DD statements for any unnecessary files.

## Deleting PDSOUT Files

### To delete PDSOUT files

1. Use the PDSOUTS member to delete PDSOUT and User Text files.
  - Uncomment the `DDISP='MOD, DELETE'` symbolic
  - Comment out the `DDISP=(null)` symbolic
2. Comment out all PDSOUT datasets except those that you want to delete. The others will remain allocated.
3. Ensure that the `ESPACE='10,5,5) ) '` symbolic is uncommented and that the `ESPACE=,` symbolic is commented out.



## Storing User Text

You can store User Text in a general-purpose library (Node1.USERTEXT.UTGP) or in a separate User Text library for any report type. This table indicates the User Text libraries, the report types they contain, and their associated ddnames:

User Text Library	Report Types	ddname
node1.USERTEXT.UTJL	Structured JCL Listing	JLJUDJL
node1.USERTEXT.UTDX	Data Set XREF	JLJUDDX
node1.USERTEXT.UTGX	Program XREF	JLJUDGX
node1.USERTEXT.UTCX	PROC XREF	JLJUDCX
node1.USERTEXT.UTIL	I/O Listing	JLJUDIL
node1.USERTEXT.UTFC	Flowcharts	JLJUDDFC
node1.USERTEXT.UTJD	Job Documentation	JLJUDJD
node1.USERTEXT.UTTU	Tape Usage Analysis	JLJUDDTU
node1.USERTEXT.UTRA	Report Analysis	JLJUDRA

The PDSOUTS JCL member is shipped with allocation statements for JLJUD (general purpose) and JLJUDJD (job documentation), as well as for a User Text Skeleton file. To allocate additional User Text libraries, add DD statements to the PDSOUTS JCL using the above table as a guideline.

## Editing DOCOPT02 File Allocations

### To specify libraries in SJJSPARM file allocations

1. Edit member DOCOPT02 in the SJJSPARM library.
2. In the File Allocations section, replace the dataset names on appropriate DD statements with the name of the datasets just allocated.
3. Comment out any unnecessary statements with an asterisk (\*) in column 1.
4. Add File Allocation statements for any additional User Text libraries that you created.

## Setting Up Optional Interfaces and Exits

This section describes how to set up optional security exit, library management, database, and graphics interfaces.

### Security Exit Interface

For information about the product security exit interface, including product access control, security exit modules, registers used by the exit program, program requirements, and the Security Interface Parameters (SIP) list passed by the product, see [Security Exit Interface](#).

## Library Management Interfaces

For information about optional interfaces that allow DOCU/TEXT to process JCL resident in CA-PANVALET library, see [Library Management Interfaces](#).

## Database Interfaces

For information about optional interfaces that allow DOCU/TEXT to process database information for IMS, DB2, and IMS/DB2, see [Database Interfaces](#).

## DB2 Table and View Cross-Reference

The procedure in this section describes how to set up DOCU/TEXT to allow cross-referencing of DB2 tables and views. Follow these instructions to enable DOCU/TEXT to read the IBM DB2 system information and store DB2 table and view references.



DOCU/TEXT includes DB2 table and view references found in DB2 plans. References from DB2 packages are not included.

### *To set up the environment*

1. Select the DB2I Primary Options Menu.
2. Select DB2I Defaults. Make changes as necessary and ensure that the correct DB2 subsystem identifier is specified.
3. Return to the DB2I Primary Options Menu and select BIND.
4. Enter the Database Resource Module (DBRM) member name `D40XTVP`. `D40XTVP` is shipped in the `SJSSAMP` library. You can copy this member to any DBRM library that you can BIND.
5. Enter the DOCU/TEXT `SJSSAMP` library name as the DBRM library.
6. Enter the plan name `D40XTVP` and change the options as necessary.
7. Specify the Run Option `TYPERUN=DB2AD` in the `DOCOPT02` member in the `SJSPARM` library. This enables DOCU/TEXT to read the IBM DB2 system information during report CREATE or ONLINE XREF maintenance process. DB2 table and view references are displayed in report DISPLAY or ONLINE XREF Query.

## Graphics Interfaces

DOCU/TEXT supports a variety of standard graphics interfaces and modes.

### IBM 3800 Line Mode Graphics

When operated in Line Mode (non-AFP), you can use the IBM 3800 Model 1 printer and the IBM 3800 Model 3 printer to generate graphics included in the Flowcharts report. This section explains how to add fonts to `SYS1.IMAGELIB` and to invoke a graphics font when creating the Flowcharts report.

### Using Graphics Fonts

Two control members generate the fonts:

- P3800 (for Model 1 printers)
- P38003 (for Model 3 printers)

If your site has both printers, you must install both members.

Only uppercase characters are supported for P3800 and P38003; therefore, you should send reports that contain both uppercase and lowercase User Text to a printer that does not support these special fonts. If the special font is not used, User Text in the Flowcharts report is translated to uppercase before printing.

Before adding fonts to SYS1.IMAGELIB, make sure the 3800 has two optional extra WCGMs. (These can be purchased from IBM.) Also, determine whether the print standard at your installation is 12-pitch or 15-pitch.

### ***Adding Fonts to SYS1.IMAGELIB***

#### ***To add fonts to SYS1.IMAGELIB***

1. Access the IEBIMAGE JCL (member P3800 or P38003) in the GRAFLIB library (unloaded from the release tape).
2. Modify the JOB statement with your site-specific parameters.
3. Submit the JCL for execution.

### ***Specifying Printer Options***

#### ***To specify printer options***

1. Add `CHARS=DTxx` to the SYSOUTs in the DT5PROC member in the SJJSSAMP library. Use DT12 for 12-pitch or DT15 for 15-pitch. (Only 15-pitch is available for Model 3 printers.) For example:

```
//JLJDOC DD SYSOUT=&SOUT.CHARS=DT15
```

2. Edit the DOCOPT02 member in the SJJSPARM library. Add a `PRINTER=3800` Run Option detail statement. For example:

```
RUN OPTIONS
PRINTER=3800
```

- If a `PRINTER=IMPACT` statement is already included as a detail statement, replace it with `PRINTER=3800`.
- If both 3800 and impact printers are used at your site, change the parameter from 3800 to `IMPACT` (or vice versa), as appropriate for the particular execution of DOCU/TEXT.

### ***Routing the Report to a Separate Output File***

An alternative to specifying printer options is to route the Flowcharts report to a separate output file that you can direct to the graphics printer.

#### ***To route the Flowcharts report to a separate output file***

1. Edit the DOCOPT02 member in the SJJSPARM library.
2. Add a DD Flowcharts Report Option detail statement and specify a ddname. For example:

```
REPORT OPTIONS
FLOWCHARTS=DD=FCDD
```

- Allocate the ddname and route it to the graphics printer. For example:

```
//FCDD DD SYSOUT=G,CHARS=DT15
```



You can also specify CHARS and PRINTER values on the Hardcopy Output Characteristics panel.

## IBM AFP Page Mode Graphics PMF Installation

When operated in Page Mode (also known as AFP mode, FSS mode or subsystem mode), you can use the IBM 3800 Model 3 printer to create graphics included in the Flowcharts report. This section explains how to install Page Mode support using the IBM Print Management Facility (ISPF dialog), and how to invoke a graphics font when creating the Flowcharts report

### Installing Page Mode Support

#### To install Page Mode support

- Select the LIBRARY SERVICES option on the PMF Main Menu.
- Select the Import Function option from the Library Services menu.
- Import these members from the GRAFLIB library (loaded at the time of installation):
  - Character group GRF2DT03
  - Character group GRF2DT04
  - Code page T1DT15BA
  - Font X1DT15



The Import Data Set Type of the input dataset for both character groups is EXP.

- Import these members from the SYS1.AIMAGE library (supplied by IBM with PMF):
  - Character group G1D0GT15
  - Character set C1D0GT15



The Import Data Set Type of the input dataset for both character groups is EXP.

- Return to the PMF Main Menu and select the Font Management Services option.
- Select the Character Set Definition option from the Font Management Services menu.
- Edit the character set C1D0GT15.
  - Add GRF2DT03 and GRF2DT04 to the list of character groups.
  - Add a comment to explain that the character set is for an ASG product.
  - Save the modified character set.

8. Within the Character Set Definition option, build character set UMDT15 from character set C1D0GT15 (edited in [step](#) ). Specify the printer device type as either 3800M3 (IBM 3800 Model 3 printer), 3820 (IBM 3820 printer), or other PMF-supported AFP printer type.

★ The PMF adds the prefix C1 for the 3800 printer or C0 for any other AFP printer when writing character set UMDT15 to the output dataset.

9. Select the Code Page Definition option from the Font Management Services menu, then build code page DT15BA from T1DT15BA imported in [step](#) .

★ The PMF adds the prefix T1 when writing code page DT15BA to the output dataset.

10. Select the Font Definition option from the Font Management Services menu, then build font DT15 from X1DT15 imported in [step](#) . Specify the printer device type as either 3800M3 (IBM Model 3 printer), 3820 (IBM 3820 printer), or other PMF-supported AFP printer type.

★ The PMF adds the prefix X1 for the 3800 printer or X0 for any other AFP printer when writing font DT15 to the output dataset.

## Specifying Printer Options

### To specify printer options

1. Add `CHARS=DTxx` to the SYSOUTs in the DT5PROC member in the SJJSSAMP library.
2. Edit the DOCOPT02 member in the SJJSPARM library. Add a `PRINTER=3800` Run Option detail statement. For example:

```
RUN  OPTIONS
PRINTER=3800
```

- If a `PRINTER=IMPACT` statement is already included as a detail statement, replace it with `PRINTER=3800`.
- If both 3800 and impact printers are used at your site, change the parameter from `3800` to `IMPACT` (or vice versa), as appropriate for the particular execution of DOCU/TEXT.

### Routing the Report to a Separate Output File

An alternative to specifying printer options is to route the Flowcharts report to a separate output file that you can direct to the graphics printer.

### To route the Flowcharts report to a separate output file

1. Edit the DOCOPT02 member in the SJJSPARM library.
2. Add a DD Flowcharts Report Option detail statement and specify a ddname. For example:

```
REPORT  OPTIONS
```

```
FLOWCHARTS=DD=FCDD
```

3. Allocate the ddname and route it to the graphics printer. For example:

```
//FCDD DD SYSOUT=G,CHARS=DT15
```



You can also specify CHARS and PRINTER values on the Hardcopy Output Characteristics panel.

## IBM AFP Page Mode Graphics non-PMF Installation

You can use the IBM AFP Page Mode printers (such as the IBM 3900 printer) to create graphics included in the Flowcharts report. This section explains how to install Page Mode support if your site does not use the IBM Print Management Facility and how to invoke a graphics font when creating the Flowcharts report. AFP printer support installation consists of installing a specialized font and associated character set and code page into the printer's font library.

### Installing Page Mode Support

#### To install Page Mode support for an IBM 3800

1. Unload the A3800LIB dataset from the product files (uncomment the A3800LIB statements in the LOADALL member in the SJJINST library).
2. Copy these members from A3800LIB to the FONTLIB(s) associated with the 3800(s) functional subsystems:
  - Character set C1UMDT15
  - Code page T1DT15BA
  - Font X1DT15

#### To install Page Mode support for any AFP printer other than an IBM 3800

1. Unload the AFPLIB dataset from the product files (uncomment the AFPLIB statements in the LOADALL member in the SJJINST library).
2. Copy these members from AFPLIB to the FONTLIB(s) associated with the printer's functional subsystems:
  - Character set C0UMDT15
  - Code page T1DT15BA
  - Font X0DT15

### Specifying Printer Options

#### To specify printer options

1. Add `CHARS=DTxx` to the SYSOUTs in the DT5PROC member in the SJJSSAMP library.
2. Edit the DOCOPT02 member in the SJJSPARM library. Add a `PRINTER=3800` Run Option detail statement. For example:

```
RUN OPTIONS
PRINTER=3800
```

- If a PRINTER=IMPACT statement is already included as a detail statement, replace it with PRINTER=3800.
- If both AFP and impact printers are used at your site, change the parameter from 3800 to IMPACT (or vice versa), as appropriate for the particular execution of DOCU/TEXT.

### ***Routing the Report to a Separate Output File***

An alternative to specifying printer options is to route the Flowcharts report to a separate output file that you can direct to the graphics printer.

#### ***To route the Flowcharts report to a separate output file***

1. Edit the DOCOPT02 member in the SJJSPARM library.
2. Add a DD Flowcharts Report Option detail statement and specify a ddname. For example:

```
REPORT OPTIONS
FLOWCHARTS=DD=FCDD
```

3. Allocate the ddname and route it to the graphics printer. For example:

```
//FCDD DD SYSOUT=G.CHARS=DT15
```



You can also specify CHARS and PRINTER values on the Hardcopy Output Characteristics panel.

### **Xerox 9700 (8700) Graphics**

This section explain how to install the special font for the Xerox 9700 (8700) printer. The special font allows you to create the graphics included in the Flowcharts report. To order the Xerox font, call or write ASG, and indicate whether the printer is online or stand-alone.

#### ***Installing the Font***

Before installing the font, confirm that you have the ASG product files containing the Xerox font.

Only uppercase characters are available in the Xerox 9700 font; therefore, you should send reports that contain both uppercase and lowercase User Text should to a printer that does not support this special font. User text in the Flowcharts report is translated to uppercase before printing if the special font is used.

#### ***Installing the Font for a Stand-alone System***

##### ***To install the font for a stand-alone system***

1. Load the font. The tape label information is:
  - LABEL=(1,NL)
  - RECFM=U
  - BLKSIZE=512
2. Put the system offline.
3. Mount the ASG tape containing the font.

4. Enter this command:

```
COPY TAPE READ LABEL ALL
```

## Installing the Font for an Online System

### To install the font for an online system

1. Load the font. The tape label information is:
  - LABEL=(1,NL)
  - RECFM=U
  - BLKSIZE=512
2. Place the printer in HOSTOCOPY mode.
3. Use JCL similar to this to route the tape contents to a Xerox printer:

---

```
//STEP00 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=XEROX,VOL=SER=XXXXXX,UNIT=TAPE,
// LABEL=(1,NL),DISP=OLD,DCB=(RECFM=FB,
// LRECL=128,BLKSIZE=512)
//SYSUT2 DD SYSOUT=your_9700_class,DCB=(RECFM=F,
// LRECL=128,BLKSIZE=128)
//SYSIN DD DUMMY
```

---

4. If 8700 graphics are being installed, enter this command:

```
COPY TAPE READ LABEL ALL
```

## After Installation

After installing the font, you must complete these steps:

1. Execute these Xerox commands:

```
FIX DSS11L.FNT
FIX DSS11P.FNT
```

If you are using Xerox Version 10 or later software, you can use this generic Xerox command instead:

```
FIX DSS*.FNT
```

2. Before using the font, complete the final steps outlined in [Establishing a JDE](#).

## Specifying Printer Options

### To specify printer options

- Edit the DOCOPT02 member in the SJJSPARM library. Add a `PRINTER=9700` Run Option detail statement. For example:

```
RUN OPTIONS
PRINTER=9700
```

- If a `PRINTER=IMPACT` statement is already included as a detail statement, replace it with



```
PRINTER=9700.
```

- If both 9700 and impact printers are used at your site, change the parameters from 9700 to IMPACT (or vice versa), as appropriate for this particular execution of DOCU/TEXT.



You can also specify the PRINTER value on the Hardcopy Output Characteristics panel.

### ***Routing the Report to a Separate Output File***

An alternative to specifying printer options is to route the Flowcharts report to a separate output file that you can direct to the graphics printer.

#### ***To route the Flowcharts report to a separate output file***

1. Edit the DOCOPT02 member in the SJJSPARM library.
2. Add a DD Flowcharts Report Option detail statement and specify a ddname. For example:

```
REPORT OPTIONS
    FLOWCHARTS=DD=FCDD
```

3. Allocate the ddname and route it to the graphics printer. For example:

```
//FCDD DD SYSOUT=G
```

### ***Establishing a JDE***

Before using the font, you must establish a JDE.

#### ***To establish a JDE to define the font***

1. Load the special font, named DSSI1L, into the Xerox laser subsystem. DSSI1L is a landscape font in ASCII format (standard for the 9700/8700). The method for invoking the font depends on your site standards and the 9700/8700 system. This is the only required parameter:

```
FONTS=(DSSI1L)
```

2. Design a DJDE to invoke the JDE. The format and content of the DJDE is site-dependent.
3. Place the DJDE in a sequential dataset with these characters:

```
RECFM=FBA
LRECL=133
BLKSIZE=6118
```

4. Add an IEBGENER step preceding any DOCU/TEXT steps in DT5PROC.

Use the DJDE dataset name as SYSUT1; use the output class for the DOCU/TEXT reports as SYSUT2. For example:

---

```
//GENER EXEC PGM=IEBGENER
//SYSPRINT DD DUMMY
//SYSUT1 DD DSN=DJDE.DSSIL,DISP=SHR
//SYSUT2 DD SYSOUT=x,DCB=BLKSIZE=133
//SYSIN DD DUMMY
```

---

5. If your site has a specific SYSOUT class assigned to the Xerox 9700, change the value of the SOUT symbolic in the DOCU/TEXT PROC (DT5PROC) to reflect that same value.

## ROSCOE Interface

ASG provides facilities that enable ROSCOE users to run the product in the ROSCOE RPF or ETSO environment. You can run the RPF and ETSO facilities independently or combined (that is, portions of the two facilities can be used simultaneously).

### RPF Installation Procedure

To install the ROSCOE RPF facility, you must complete these tasks:

- Execute member DTRUNLD.
- Modify member DTRMAIN.
- Review/modify member DTR11.

#### Step 1—Execute Member DTRUNLD

##### To execute member DTRUNLD

- a. Copy member DTRUNLD from the SJJSRPF library and save it into a ROSCOE library suitable for RPF execution. DTRUNLD is an RPF, written to simplify the RPF and panel installation.
- b. Execute member DTRUNLD from the new location.
  - At the first prompt, enter the fully-qualified dataset name `&DJSNAME.SJJSRPF` unloaded from the product files.
  - At the second prompt, enter the two- or three-character prefix of the ROSCOE library from which these RPFs will execute.  
  
ASG recommends that you place these RPFs in a GLOBAL or system library that is accessible to all users.
  - Ensure that the ROSCOE ID has sufficient authority to update this library.

#### Step 2—Modify Member DTRMAIN

##### To modify member DTRMAIN

- a. ATTach member DTRMAIN and change the file name assigned to permanent variable P16 to correspond to the high-level qualifiers of the product libraries from the download (do not modify the last node).
- b. Review permanent variable P14 on line 19 for the appropriate value at your site. The valid options are `STD`, `ALL`, `AD`, and `OD`. In most cases, `STD` is the preferred option.
- c. After making the necessary changes, resave the updated RPF.

#### Step 3—Review/Modify Member DTR11

If your site intends to implement the User Text feature using PDSs or Librarian files for text storage, this step is required. When you use the User Text Maintenance panel to CREATE a new text member, a BLNKFORM member from the User Text library (a skeleton form) is imported to your AWS and ATTached to it.

- If there is no member by that name in that library, the RPF displays a `MEMBERNOTFOUND` message in the upper right corner of the panel.
- If you want a different name of the skeleton form, change the value of permanent variable `P15` on line 14 of RPF `DTR11` to reflect the desired name.

## RPF Installation Panels

The following panels are typically displayed during the installation and initial execution of `DOCU/TEXT` under the `ROSCOE` RPF.

```

          DOCU/TEXT RPF UNLOAD PANEL

Enter the data set name into which the DOCU/TEXT RPFs were unloaded:

Enter the ROSCOE PREFIX into which the RPFs are to be copied:
===>

PRESS PF3/PF15 to terminate without RPF execution.

```

```

*****
*
*   This is the first execution of DOCU/TEXT under your USERID.
*
*   DOCU/TEXT performs many functions which require batch execution.
*   Other functions require selection of job JCL from JCL libraries.
*   The next two panels you will see require information which
*   DOCU/TEXT will need to correctly perform these functions in a
*   ROSCOE environment.
*
*   When the entries are complete, the regular DOCU/TEXT menu will be
*   presented.
*
*   Press ENTER to move on to the panels.
*   Press PF3/PF15 to exit and return to ROSCOE command mode.
*
*****

```

## ETSO Installation Procedure

Prior to installing the ETSO facility, complete the regular product installation. If you do not intend to use the ISPF panels, the `SJJSPLIB`, `SJJSMLIB`, and `SJJSCLST` libraries are not required. The `SJJSCLST` library is not used in any case.



The `DOCU/TEXT` ISPF panels utilize PDF as well as ISPF. You must have the PDF extension of `TSO/ISPF` to take full advantage of the `DOCU/TEXT` ISPF panels.

To install the `ROSCOE` ETSO facility, you must complete these tasks:

- Establish the ETSO environment using member `DTETSOPG` or `DTETSOP7`.
- Modify member `DTRISPF`.

## Step 1—Establish the ETSO Environment

### To establish the ETSO environment

- a. Copy member DTETSOPG or DTETSOP7 from the `xxxxxx.xxxxxx.SJJSRTSO` library and merge one of these entries into the ETSOPGMS member in your main ROSCOE library (usually the RO prefix).
- b. Verify that the entries are in alphanumeric sequence and replace the ETSOPGMS in the main ROSCOE library. This table lists the ETSOPGMS entries:

Entry	Function
ISPSTART	ISPF panels
D00YDMU	DOCU/TEXT panels
D70YCMU	ISPF panels

- c. If you plan to use the ISPF panels, you should install the member CLLEXIT (ACF2 Exit) into your ROSCOE.SOURCE library. This allows ROSCOE to return a usable user ID for an ISPF dialog. (CA/ROSCOE support can assist you further.)
- d. If your site security package is ACF2, you must define all of the program name entries in ETSOPGMS into the ACF\$RCLL member in ACF2.MACLIB (site dependent). You should assemble this member prior to any ETSO execution. This enables ROSCOE to execute the programs referenced in ETSOPGMS interactively.
- e. Add these ROSCOE initialization parameters to your ROSCOE startup parms:

Parameter	Description
ESCAPE=.	Escape sequence.
ETSALLOC=45	Max dynamic file allocation.
ETSOPGMS=RO.ETSOPGMS	Eligible program list.
ETSSLICE=3000	Time slice default.
ETSSRCH=ALL	Optional. Called PGMs (ETSOLIB, JOBLIB, LPA, and LLA). If you specify ETSSRCH=ALL, you can add your DOCU/TEXT load library to the Link Pack Area (LPA), Link List Area (LLA), JOBLIB, STEPLIB, or ETSO DD statement. If you specify ETSSRCH=DD, you must concatenate the product load library in the ETSO library DD statement. Refer to the <i>ROSCOE System Reference</i> for ETSO

Parameter	Description
	application installation details.
PSCBUSER= <i>aaaa</i>	Pseudo UID.

### Step 2—Modify Member DTRISPF

Perform these steps only if you plan to use the ISPF panels via ROSCOE.

#### To modify member DTRISPF

- Copy member DTRISPF from the `xxxxx.xxxxx.SJSRTSO` library. Modify line 3. Change `xxxxx.xxxxx` to the high-level qualifier used when downloading the product files.
- Locate the line containing `ALLOC ISPPROF`. Change the `YOUR.` dataset name to reflect a naming convention compatible with your site standards.

The dataset should use a high-level qualifier that is accessible to all users for create and update functions. ASG recommends that you set the second node to your ROSCOE prefix. The dataset must be a PDS with the attributes of `RECFM=FB` and `LRECL=80`. Five tracks should be adequate.



If the DTRISPF RPF does not build the dataset for you, it must be pre-allocated by/for each user prior to execution.

- Locate the lines that contain the string `YOUR.` and modify each one to reflect the appropriate IBM shipped ISPF/PDF library. It is not necessary, nor desirable, to identify any other vendor libraries.
- Save DTRISPF into a universally accessible ROSCOE library as `DOCUTEXT` or any other desirable name.

The ROSCOE installation is now complete.

## Final Installation Tasks

Perform these final tasks after you have completed all other installation tasks:

- Initialize DOCU/TEXT. Follow the procedures outlined in the topic [Initializing DOCU/TEXT in ISPF in ASG-DOCU/TEXT's System Administration guide](#).
- Initialize the SDF/OLX file (if you created one).

The SDF/OLX file stores Short Descriptions and cross-reference information available through the OLX Query facility. To initialize the file, follow the instructions in [User Information - JOB/SCAN](#) to update the file with at least one job or Short Description.

If an allocation error occurs, confirm that:

- The dataset name coded in `DOCOPT02` was spelled correctly.
- The file was created properly.

- You are authorized to view and update this file.

## Calling JOB/SCAN from a User Program

If you plan to invoke JOB/SCAN from a user program, take these considerations into account:

- JOB/SCAN is callable from any user program using standard procedures (that is, all files that JOB/SCAN needs should be allocated prior to the call). Refer to the *Validating JCL in Batch* for batch execution JCL samples showing program and ddnames.
- JOB/SCAN is not re-usable. Therefore, repeated invocations from a main program should be done either using the LINK macro (this is the preferred method), or by a LOAD, BRANCH and DELETE sequence.
- JOB/SCAN reads the JCL to be scanned from a file, which is dynamically allocated to ddname JIJOB. It opens, reads, and then closes the file.
- JOB/SCAN places its output listing in a file with a ddname of JLKAC.
- JOB/SCAN expects a certain sequence of control statements in the input stream (INPUT SELECTION). These statements define what type of input should be read and what outputs to produce. These statements are documented in the section of the *Validating JCL in Batch* that describes the execution of JOB/SCAN as a batch job.
- JOB/SCAN expects to be called with a 1-byte parm field indicating the type of processing to be performed. The valid values are:

S—scan to validate

R—reformat

C—change

This example shows a sample assembler program calling JOB/SCAN:

### Assembler to call JOB/SCAN from a User program

```

USERPGM    CSECT
           ...
           ...
           ...
           LINK    EP=J02YJS,PARAM,VL=1
           LTR     R15,R15
           ...
           ...
           ...
JSPARM     DC    H'1'
           DC    CL1'S      '

```

# System Administrator Tasks

This documentation provides complete information for the System Administrator. The document provides information on the libraries used by ASG-JOB/SCAN (herein called JOB/SCAN), describes how to use control statements, File Allocation statements, and Control Tables, and explains numerous Run and Report Options statements.

Topic	Description
<a href="#">Administration Basics</a>	Introduces Product Administrator functions and the Product Administration panel.
<a href="#">Administering JOB/SCAN</a>	Explains how to perform the key JOB/SCAN administrative tasks.
<a href="#">Options</a>	Describes administrative runtime options.
<a href="#">JOB/SCAN Libraries</a>	Introduces SJJSPARM library members and explains how to create and modify SJJSPARM members. Provides information about the product libraries: SJJSSAMP, SJJSTDL, and SJJSDemo.
<a href="#">Product Usage Facility</a>	Describes administration and use of the product usage facility.

---



# Administration Basics

This section describes the basic administrator's role and duties.

- [System Administrator Role](#)
- [Using the Product Administration Panel](#)
- [Options Overview](#)

## System Administrator Role

The JOB/SCAN System Administrator (herein called the Administrator) selects the options and features of JOB/SCAN to meet site, group, and user requirements, maintains standardized usage, and authorizes users' and groups' access to the product.

This table shows where the feature or function is described in this or another JOB/SCAN documentation.

Function or Feature	Reference
Procedures for administering, configuring, and authorizing use of JOB/SCAN	<a href="#">Administering JOB/SCAN</a>
Operational control	<a href="#">Options</a> , with complete details in <a href="#">Reference Information</a>
SJJSPARM and other JOB/SCAN libraries	<a href="#">JOB/SCAN Libraries</a>
Input Selection Statements	<a href="#">Input Selection Group</a>
Exits	<a href="#">Setting Up Optional Interfaces and Exits</a>

## Using the Product Administration Panel

The Product Administration panel contains the list of administration tasks.

### Displaying the Product Administration panel

#### *To display the Product Administration panel*

- Select the Product Administration option from the ASG-JOB/SCAN Primary Menu.

JOB/SCAN displays the Product Administration panel.

**Product Administration Panel**

```

----- PRODUCT ADMINISTRATION -----
COMMAND  ===>

ASG Customer Number NNNNNNNNN

  0 - Verify Access for USERID
  1 - Change System Identification and Access Control
  2 - Modify Options Members
  3 - Test An Options Member
  4 - Display CPU Serial Number
  5 - Product Usage Administration

ASG-JOB/SCAN product authorization code will expire on 20nn-12-31

```

This table describes the options available from this panel.

Menu Option	Purpose
Verify Access for USERID	View the default or assigned options currently available for each selected userid or groupid. (See <a href="#">Verifying User Access</a> )
Change System Identification and Access Control	Invoke ISPF Edit on SJJSPARM member JSOPT01 to change the System Identification or other group and detail statements. (See <a href="#">Modifying and Testing SJJSPARM Members</a> )
Modify Options Members	Display a directory of the entire SJJSPARM library. Select a member for modification and then modify it in ISPF Edit. (See <a href="#">Administering JOB/SCAN</a> )
Test an Options Member	Perform a syntax validation in foreground on any Options Member. (See <a href="#">Administering JOB/SCAN</a> )
Display CPU Serial Number	Display the serial number of the CPU being used. (This number is a factor in the Product Authorization code.) (See <a href="#">Displaying the CPU Serial Number</a> )
Product Usage Administration	Maintain the usage statistics for JOB/SCAN. Grant access to users to view usage data. Export usage data to a file suitable for manipulation by spreadsheets and other programs. (See <a href="#">Managing Database Interfaces</a> )

**Displaying the CPU Serial Number**

Because the JOB/SCAN Product Authorization code is based on the CPU serial number, ASG customer support might, in certain circumstances, request the serial number.

**To display the serial number of the CPU being used**

- From the Product Administration panel, select Display CPU Serial Number option. JOB/SCAN displays the Display CPU Serial Number(s) panel.

### Display CPU Serial Number(s) Panel

```

----- DISPLAY CPU SERIAL NUMBER(S) -----
COMMAND ==>

      CPU Model/Type           ==> 2817/407
      CPU Serial Number        ==> 0546B6
      Second Serial Number     ==>
      Product Release          ==> JS7.6.00

```

### Verifying User Access

**To verify the default or assigned options currently available for selected userids and groupids**

1. Select Verify Access for USERID from the Product Administration panel. JOB/SCAN displays the Verify Access Control for USERID panel.

### Verify Access Control for USERID Panel

```

----- VERIFY ACCESS CONTROL FOR USERID -----
COMMAND ==>

      User ID to be verified ==>
      Group ID                ==>

```

2. Type a userid or groupid, and press Enter. JOB/SCAN displays a list showing the functions and facilities available to the user.
3. Verify the user's access and change as appropriate.

### Verify Access for USERID Panel

```

----- VERIFY ACCESS FOR USERID -----
COMMAND ==>
User ID => ASGUSR
Group ID =>

```

	Access	Options
Initialization		=> N
Execute JOB/SCAN	=> Y	=> N
Selection List Maintenance	=> Y	
Assign Alternate Options Members	=> Y	
Product Administration	=> Y	
JOB/SCAN Usage Report	=> Y	
Standards Program/Parm	=> ANY	=> ANY
Standards Prompter	=> Y	

The Verify Access for USERID panel allows the Product Administrator to designate which facilities are accessible to a specific userid, and whether alternate SJJSPARM library options members are assigned.

## Options Overview

Options statements control, regulate, and modify all aspects of JOB/SCAN processes, including:

- authorization of the many product features
- titles and contents of the reports
- how the JCL statements and their parameters get validated. [Reference Information](#) describes each option in detail.

SJJSPARM library members contain options statements.

Put related options statements (called *detail* statements) together under their assigned *group* statement.

### Group-Level Statements

When specifying options in detail-level statements, begin with the group-level statement and follow it with as many detail statements as necessary. The statement group can appear anywhere in the SJJSPARM library member, provided the group-level statement immediately precedes the detail statements. Include multiple group-level statements within an SJJSPARM library member. The group-level statements are:

Group-Level Statement	Function
System Identification	Contains the TITLE1 value for all JOB/SCAN reports.
Access Control	Allows control of user access to panels from the ASG-JOB/SCAN Primary Menu panel.
File Allocations	Assigns work files and user PROCLIBs.
Run Options	Controls overall JOB/SCAN processing, including access to VTOCs, control card libraries, system catalogs, and environment-specific processing.
Report Options	Controls the reports produced.
Control Tables	Used in conjunction with Run Options to define the JOB/SCAN environment and Reformatter settings.
Input Selection	Designates libraries and members to be selected for processing. Input Selection statements need to be coded only when running in batch mode.

### Detail-Level Statements Format Rules

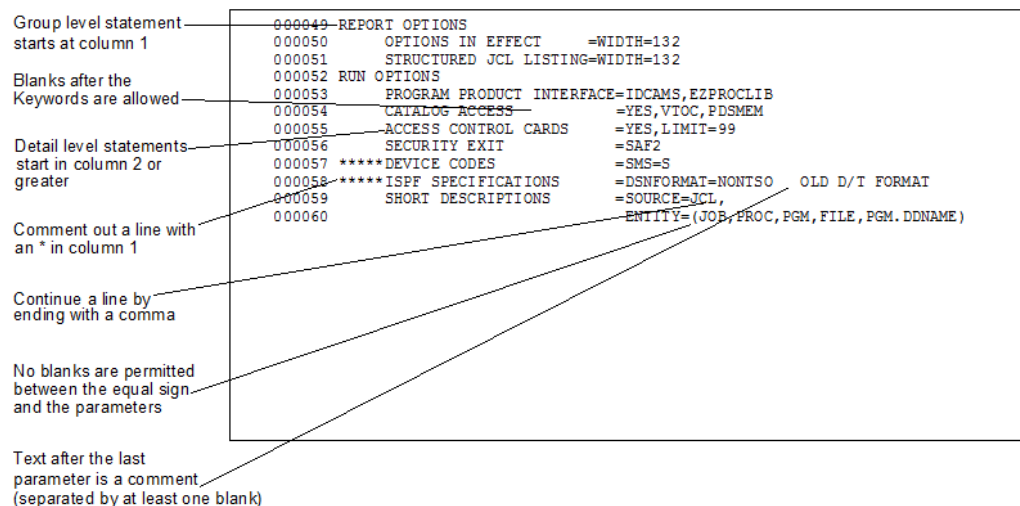
Subordinate to each group-level statement are the detail statements containing various parameters for the group.

Follow these formatting rules when entering detail-level statements:

- Detail-level statements can appear in any order. Certain detail statements and parameters can be repeated depending on the statement and can have a cumulative or restrictive effect.
- An acronym can replace the detail statement keyword. In this documentation, the topic headings for each run option include the option's acronym in parentheses; for example, (ACC).
- The detail statement keyword is followed by an equal (=) sign. Spaces between the end of the keyword and the equal sign are allowed.

- The parameters follow the equal (=) sign with no intervening blanks allowed.
- Statement parameters are optional and may be entered in any order. Multiple parameters must be separated with commas.
- Continue a detail statement to the next line by ending the line with a comma and putting the next parameter on the following line in column 2 or greater.
- Comments can be placed on any line following a blank after the last parameter.
- Comment out a line with an asterisk (\*) in column 1.

### Options Member highlights



# Administering JOB/SCAN

After JOB/SCAN installation, the Administrator performs these tasks:

- [Assigning the Product Administrator Role](#)
- [Controlling Access to JOB/SCAN](#)
- [Managing JOB/SCAN through Options Members](#)
- [Supplementing SJJSPARM Options](#)
- [Allowing Users to Create Lists of Jobs](#)
- [Allowing Users to Create Standards Programs](#)
- [Adding Items to the Standards Utility Menu Panel](#)
- [Increasing File Space Allocation](#)
- [Adding Site-Defined Functions](#)
- [Managing Database Interfaces](#)
- [Managing Scheduler Interfaces](#)
- [Managing Security Interfaces](#)
- [Optimizing JOB/SCAN's Resource Requirements](#)
- [Ensuring Equivalent JCL Validation Results](#)
- [Managing JOB/SCAN's Output Messages](#)

## Assigning the Product Administrator Role

The Administrator detail statement assigns the Administrator role in the JSOPT01 member in the SJJSPARM library. This member name cannot be overridden. It controls user access to the Product Administration panel by including an entry on the ASG-JOB/SCAN Primary Menu panel. See [Administrator Detail Statement](#) for syntax and details about the Administrator Statement and [Using Access Control Detail Statements](#) for details about options set up and usage.



Because access to the Administration panel grants users complete access to JOB/SCAN, ASG recommends that access to the Product Administration panel be restricted to those individuals requiring administration credentials.

### Example 1

This example assigns administrator privileges to one userid.

```
ADMINISTRATOR=USERID=ASGADM1
```

## Example 2

This example has lines that assign the administrator role to a single user, a pair of users, and similarly named userids, respectively.

```
ADMINISTRATOR=USERID=ASGUSR1
ADMINISTRATOR=USERID=(ASGBOSS,ASGSUPR2)
ADMINISTRATOR=USERID=ASGADM*
```

The figure shows member JSOPT01 a single userid in the administrator role.

### Modified JSOPT01 SJJSPARM member

```
EDIT          COM2.SJJSPARM(JSOPT01) - ----- Columns 00001 00072
Command ==>                                     Scroll ==> PAGE
000001 ** JSOPT01  ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** *
000002 ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** *
000003 ** THIS MEMBER ENABLES ACCESS TO JOB/SCAN AND **
000004 ** CONTROLS USER ACCESS TO PANELS AND EXTENDED FUNCTIONS. **
000005 ** ** ** *
000006 ** ** ** *
000007 SYSTEM IDENTIFICATION
000008 TITLE1='WIDGET MANUFACTURING INC'
000009 *
000010 ACCESS CONTROL
000014 ADMINISTRATOR =USERID=ASGADM01
```

## Controlling Access to JOB/SCAN

User access to JOB/SCAN is controlled by the JOB/SCAN detail statement in the JSOPT01 member in the SJJSPARM library. This member name cannot be overridden. It controls all users' access to the ASG-JOB/SCAN Primary Menu panel. This statement also grants privileges to use the Standards

Facility, the Reformatter command language, and the Submit After Checking option on the JOB/SCAN Validation panels. See [JOB/SCAN Detail Statement](#) for syntax and details about usage. See [Using Access Control Detail Statements](#) for details about using multiple detail statements.



ASG recommends that write access to the SJJSPARM library be restricted to administrators.

### Example 1

This example allows all the users to use JOB/SCAN, use any Standards and any parameter for the Standards, and for JOB/SCAN to submit jobs after validation.

```
JOB/SCAN=ALL, STANDARDS=ANY, PARM=ANY, SUBMIT
```

### Example 2

This example allows two users, ASGUSR1 and ASGUSR2, to use JOB/SCAN and use any Standards program. Also, the example restricts the rest of the users to use only JOBCHK standards and to submit jobs after validation.





## Managing JOB/SCAN through Options Members

JOB/SCAN options come from two types of sources: options members (default and alternate) and ISPF panels. Options members are usually kept in the SJJSPARM dataset. Options from ISPF panels come from the Validation and the Processing Options panels.

### Default Options Members

The SJJSPARM library contains *options members* that JOB/SCAN reads during execution.

JOB/SCAN concatenates the SJJSPARM options members that it uses for JCL Validation to the ddname JLRCL (see the figure for order). The default options members in the SJJSPARM library are:

**JSOPT01** Contains System Identification and Access Control statements. It may also contain other option statements. JOB/SCAN reads this member first to determine which JOB/SCAN facilities or alternate members users are authorized to access. The Access Control statements in JSOPT01 determine the configuration of the user's ASG-JOB/SCAN Primary Menu panel and access to the administration panel. The name of this member cannot be changed nor substituted; only the contents of the member can be changed.

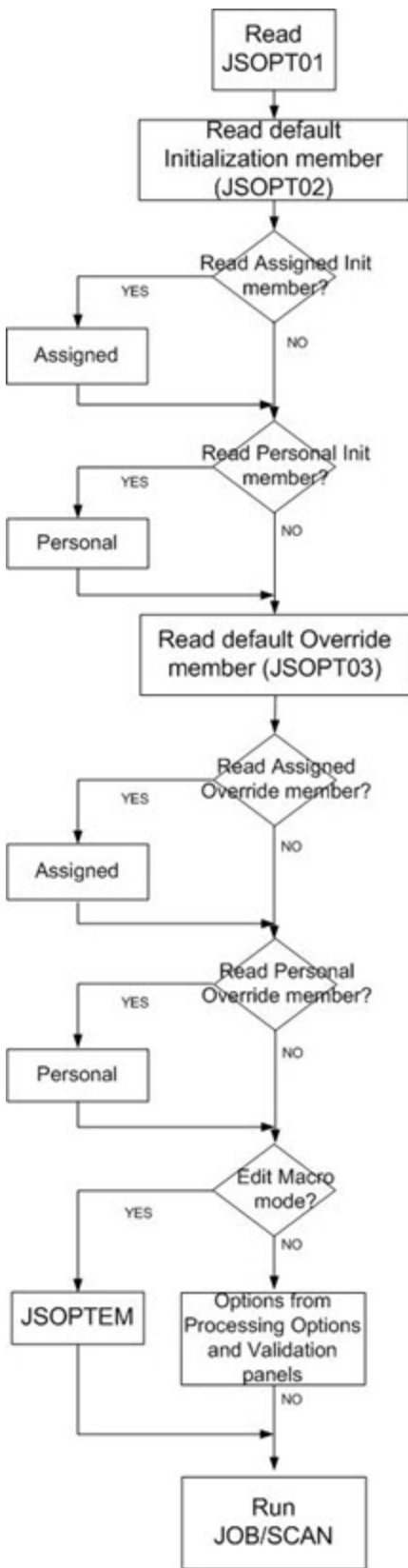
**JSOPT02** (or equivalent) Contains the File Allocations control statements required for *initialization* when the Validate JCL option is chosen from the ASG-JOB/SCAN Primary Menu panel. Typically, it contains most of the File Allocations, Run Options, and Report Options statements used when JOB/SCAN is invoked from the JOB/SCAN - JCL Validation panel and when JOB/SCAN Edit macros are used. Can also contain user-supplied Control Table statements for the Standards Facility.

**JSOPT03** (or equivalent) Contains additional control statement *overrides* used when JOB/SCAN JCL Validation is invoked from the JOB/SCAN - JCL Validation panel and from JOB/SCAN Edit macros. JSOPT03 is concatenated to ddname JLRCL following JSOPT02, and overrides conflicting statements in JSOPT02.

In foreground mode, the options entered in the JOB/SCAN Processing Options and the JOB/SCAN - JCL Validation panels override any conflicting options read from the SJJSPARM library options members JSOPT01, JSOPT02, or JSOPT03 or their equivalents. In batch mode, JOB/SCAN considers only options from these three members and control statements in the JCL for the batch run.

**JSOPTM** (or equivalent) Contains Input Selection, Run Options, and Report Options statements used by JOB/SCAN Edit macros. Use this member to specify the Standards Facility the Edit macros use. JSOPTM is used only when JCL Validation is invoked by JOB/SCAN Edit macros. JSOPTM is concatenated to ddname JLRCL following JSOPT02 and JSOPT03 and overrides conflicting statements in JSOPT02 and JSOPT03.

## Options Source precedence





JOB/SCAN can create a JSOPTJ2 options member to use, in addition to the other options members, to provide default File Allocations statements for ddname JLJPA. (See [Generating Default JLJPA Specifications](#))

## Control Table Precedence

For Control Table entries in SJJSPARM library members, the *first* value encountered for a Control Table in the SJJSPARM concatenation takes precedence over subsequent, conflicting values for the same Control Table.

## Alternate SJJSPARM Members

In place of the default SJJSPARM library options members JSOPT02 and JSOPT03, JOB/SCAN can be set up to use alternate options members: either *assigned* or *personal*. Assigned options members contain options that a user or group of users must use; this member name cannot be changed by the user. Personal members contain options users may use but are not required to use. If some users or groups intermittently need different options and control statements, create personal options to suit their needs. (See [Personal Options Members](#)).

## Assigned Options Members

The administrator creates options members and assigns them for individual or group use in place of using default options members. For example, unique file allocations can be specified for a group to use. The member is customized, given a unique name in the SJJSPARM library, and assigned to a userid or groupid.

### ***To create an assigned options member***

1. Select Modify Options Members from the Product Administration panel.  
A directory of the SJJSPARM library appears.
2. Using ISPF functions, create a new member and edit it.
3. Code the desired control statements in the new member, then press End to return to the Product Administration panel.
4. Test the new options member. See [Testing an Options Member](#) to test the new member.

Multiple, uniquely named versions of JSOPT02 and JSOPT03 allow the administrator to meet the needs of different users, user groups, or other execution requirements of JOB/SCAN.

If an alternate options member is for initialization in place of JSOPT02 or JSOPT03, the user must exit and re-enter JOB/SCAN to use the alternate member. The same options member can be used by any number of userids or groups.

### ***To assign an options member to a user or group of users***

1. Select Change System Identification and Access Control from the Product Administration panel to bring JSOPT01 into Edit.
2. For each user or group for whom you assign an options member, code the `MEMBER` parameter with the options member name in the appropriate detail statement:
  - If the options member is to be used instead of JSOPT02, code the `MEMBER=` parameter in the

Initialization detail statement for the appropriate userid or groupid (see [Initialization Detail Statement](#)).

- If the options member is to be used instead of JSOPT03, code the MEMBER= parameter in the JOB/SCAN detail statement for the appropriate userid or groupid. (see [JOB/SCAN Detail Statement](#)).

## Personal Options Members

Some users need individualized options members. The administrator creates the options member with customized options, authorizes user to use it, and users assign it to themselves by selecting Assign Alternate Options Member from the ASG-JOB/SCAN Primary Menu. Personal options members take precedence over assigned and default options members.

### *To create a personal options member*

1. Select Modify Options Members from the Product Administration panel. ISPF displays the directory of the SJJSPARM library.
2. Using ISPF functions create a new member and bring it into Edit.
3. Code the desired control statements. Press End to return to the SJJSPARM library directory. Press End to return to the Product Administration panel.
4. To test the new options member. See [Testing an Options Member](#).
5. Authorize the user to specify the use of the new options member. See [To allow users to designate personal options members](#).

Give users the ability to designate a personal options member through the Options statement in the JSOPT01 member in the SJJSPARM library (see [Options Detail Statement](#)). This statement controls user access to the Assign Alternate Options Members panel from an entry on the ASG-JOB/SCAN Primary Menu panel. Users can then designate personal options members and display default options member names and any alternate options members assigned.

### *To allow users to designate personal options members*

- Example 1. This example allows no users to assign options members.

```
OPTIONS=NONE
```

- Example 2. This example allows only two users to assign personal options members. See [Options Detail Statement](#) for syntax and details about usage.

```
OPTIONS=USERID= (ASGUSR1, ASGUSR2)
OPTIONS=NONE
```

Users that can assign themselves personal options have write access to the Personal fields of this panel:

**Modified JSOPT01 SJJSPARM member**

```

-----  ASSIGN ALTERNATE OPTIONS MEMBERS  -----
COMMAND ===>

CHANGE THE OPTIONS MEMBERS AS NEEDED:

      Initialization      Personal      Assigned      Default
      Override           ===> MYOPTMR  ===> MYOPT02  ===> JSOPT02
                        ===>                ===> MYOPT03  ===> JSOPT03

```

Users may type member names only in the Personal fields for Initialization and Override.

**Modifying and Testing SJJSPARM Members**

Options on the Product Administration panel allow the JOB/SCAN Administrator to modify the SJJSPARM library members that JOB/SCAN uses and to test the members for syntax errors after they are changed. The Change System Identification and Access Control option is used to change TITLE1 or Access Control statements in SJJSPARM library member JSOPT01. The Modify Options Members option is used to modify File Allocations statements, Control Table entries, Run Options, and Report Options in any SJJSPARM library member. The Test an Options Member option is used to test any SJJSPARM library options member for syntax errors.



Changes to SJJSPARM options will not take effect until JOB/SCAN has been initialized with the new values. After modifying SJJSPARM library members, exit and re-start JOB/SCAN.

**Partial Product Administration Panel**

```

-----  PRODUCT ADMINISTRATION  -----
COMMAND ===>

0 - Verify Access for USERID
1 - Change System Identification and Access control
2 - Modify Options Members
3 - Test An Options Member
4 - Display CPU Serial Number

```

**Modifying SJJSPARM Members*****To change options members***

1. Select the Modify Options Members option from the Product Administration panel. A list of members in the SJJSPARM library displays.
2. Select the member to be modified from the member selection list. The selected member displays in ISPF Edit mode.
3. Code the desired changes and press End twice to return to the Product Administration panel.
4. Test the member. See [Testing an Options Member](#) to test the new member.
5. Exit and re-invoke JOB/SCAN to initialize JOB/SCAN with the changed option controls.







## Sample JOB/SCAN - JCL Validation Panel

```

----- JOB/SCAN - JCL VALIDATION -----
COMMAND ==>

      F - Foreground execution mode      P - Processing Options
      B - Batch execution mode           E - Edit JCL member
                                      BR - Browse JCL member

Input Library  ==> SYSUSR.JCL
Input Type    ==> J                    (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ==> PDS                  (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> LEDGR0              (Single member if BATCH with TRANSMIT)
Exclude Members ==>                    (Not valid for BATCH with TRANSMIT)
First Proclib ==>
Proclib Type  ==>                      (PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
Report Heading ==>
Display        ==> Y (Yes/No/ErrorsOnly)  Browse/View ==> B (B/V)
Hardcopy       ==> N (Yes/No/ErrorsOnly/SpaceOnly/JobErrors)
Options in Effect Listing ==> N (Y/N)      SJL PDSOUT ==> N (Y/N)

Selection Exit ==>                      Parm ==>
Standards Name ==>                      Parm ==>

```

This table shows the correspondence between the panel fields shown in the above figure and SJJSPARM options statement parameters. (Default values are indicated by **bold type**.)

Panel Field	SJJSPARM Statement and Parameter
DISPLAY	STRUCTURED JCL DISPLAY = <b>YES</b>  NO ERRORONLY
BROWSE/VIEW	STRUCTURED JCL DISPLAY = <b>BROWSE</b>  VIEW
HARDCOPY	STRUCTURED JCL LISTING =YES  <b>NO</b>  ERRORONLY SPACEONLY JOBERRORS
OPTIONS IN EFFECT LISTING	OPTIONS IN EFFECT =YES  <b>NO</b>
LIBRARY TYPE	JLJPA PROCLIB TYPE = <b>PDS</b> /PAN/LIB/LIB-3/SEQ
PROCLIB TYPE	JLJPC PROCLIB TYPE = <b>PDS</b> /PAN/LIB/LIB-3
REPORT HEADING	REPORT HEADING ='report heading'
STANDARDS NAME and PARM	STANDARDS NAME =prgname,'parm'

### Relating Processing Options Fields to SJJSPARM Options

Report Option and Run Option parameters from the JOB/SCAN Processing Options panel override the corresponding parameters from SJJSPARM library options members when running JOB/SCAN in foreground mode.

## Sample JOB/SCAN Processing Options Panel

```

----- JOB/SCAN PROCESSING OPTIONS -----
COMMAND ==>

Use Error Table           ==> N           (Y/N)
Condition Code Processing ==> N           (Y/N)
Catalog Access           ==> Y           (Y/N)
VTOC Access              ==> Y           (Y/N)
PDS Member Verification  ==> Y           (Y/N)
Security Exit DSNAME Check ==> Y           (Y/N)
Program Checking         ==> Y           (Y/N)
Disable SMS Processing   ==> N           (Y/N)
IDCAMS Processing        ==> Y           (Y/N)
Use SCHEDULER interface  ==> Y           (Y/N)
Job Limit                ==>           (1 to 999 or Blank For No Limit)
Access Control Cards     ==> Y           (Y/N)
Control Card Limit       ==>           (1 to 999 or Blank For No Limit)
Subsystem                ==> JES2       (JES2/JES3/None)
Suppress Error Messages  ==> NO        (A,W,E,0 to 99,or NO)
Extended Error Text      ==> S           (X=Extended,S=Short)

Press <ENTER> to verify option values or <END> to leave and save option values.

```

This table shows equivalent JOB/SCAN panel fields and SJJSPARM statements for the Report Option and Run Option fields on the JOB/SCAN Processing Options panel (figure above). (Default values are indicated by **bold** type.)

Panel Field	SJJSPARM Statement and Parameter
ACCESS CONTROL CARDS	ACCESS CONTROL CARDS = <b>YES</b> /NO
CATALOG ACCESS	CATALOG ACCESS = <b>YES</b> /NO
CONDITION CODE PROCESSING	PROCESSING SPECIFICATIONS = CONDPCODE/ <b>NOCONDPCODE</b>
CONTROL CARD LIMIT	ACCESS CONTROL CARDS =LIMIT=n
DISABLE SMS PROCESSING	SMS =ACS/NOACS,CAS/NOCAS
IDCAMS PROCESSING	PROGRAM PRODUCT INTERFACE = IDCAMS/ <b>NOIDCAMS</b>
USE SCHEDULER INTERFACE	See the appropriate <i>JOB/SCAN Scheduler documentation</i> for details about this statement - <a href="#">User Information - JOB/SCAN with Schedulers</a> .
JOB LIMIT	JOB LIMIT = (x,y) where x is the batch limit and y is the foreground limit and y = 1-999. Default is no limit (Job Limit field entry overrides y from SJJSPARM only if it is less than y.)
PDS MEMBER VERIFICATION	CATALOG ACCESS = <b>PDSMEM</b> /NOPDSMEM

Panel Field	SJJSPARM Statement and Parameter
PROGRAM CHECKING	CATALOG ACCESS = <b>PGM</b> /NOPGM
SECURITY EXIT DSNAME CHECK	SECURITY EXIT =, DSNAME/ <b>NODSNAME</b>
SUBSYSTEM	SUBSYSTEM = <b>JES2</b> /JES3/NONE
SUPPRESS ERROR MESSAGES	SUPPRESS ERROR MESSAGES <b>NO</b> /A/W/E/ <i>nn</i> where <i>nn</i> = 0 - 99 is the Severity Level
USE ERROR TABLE	USE ERROR TABLE =YES/ <b>NO</b>
VTOC ACCESS	CATALOG ACCESS = <b>VTOC</b> /NOVTOC
EXTENDED ERROR TEXT	STRUCTURED JCL LISTING=ERRORTXT= <b>SHORT</b> /EXTENDED

## Allowing Users to Create Lists of Jobs

JOB/SCAN's list facility makes it easy to process members as a group through the Select Lists detail statement in the JSOPT01 member in the SJJSPARM library. The statement controls user access to the Selection List Maintenance Facility panel by including an entry on the ASG-JOB/SCAN Primary Menu panel. This facility allows users to create lists of jobs to be processed as a group.

### Example

This example allows all the userids beginning with APL to use selection lists.

```
SELECT LISTS=USERID=APL*****
```

See [Select Lists Detail Statement](#) for syntax and details about usage. See [Using Access Control Detail Statements](#) for details about using multiple detail statements.

## Allowing Users to Create Standards Programs

JOB/SCAN's standards facility speeds the development of site-specific standards programs in COBOL through the Standards Prompter detail statement in the JSOPT01 member in the SJJSPARM library. It controls user access to the Standards Utility Menu panel from the ASG-JOB/SCAN Primary Menu panel.

To facilitate access to frequently used utilities, add them to the Standards Utility Menu panel. See [Adding Items to the Standards Utility Menu Panel](#).

### Example

This example allows all the userids beginning with ASG to use the Prompter. See [Standards Prompter Detail Statement](#) for syntax and details about usage. See [Using Access Control Detail Statements](#) for details about using multiple detail statements.

STANDARDS PROMPTER=USERID=ASG\*\*\*\*\*

## Adding Items to the Standards Utility Menu Panel

Additional selection options can be added to the Standards Utility Menu. This feature provides access to functions that are not part of the product but which need to be made available from the Standards Utility Menu panel. The example shows how to include an option to access SDSF to view held output.



If the region requirements of any added function exceed 1 MB, it may be necessary to increase the overall region size beyond the normal used for JOB/SCAN.

### To add the SDSF Output Option to the Standards Utility Menu

1. Exit JOB/SCAN.
2. Make a backup copy of member J06PSTUT in the SJJSP LIB library member - call it SJJSP LIB(X06PSTUT).
3. Edit SJJSP LIB member J06PSTUT.
4. Determine the position of the new item to be added.
5. Add a blank line; then add:

```
O+ Output - View held output using SDSF
```

6. Add the selection code ("O" for Output) to the verify list by changing:

```
VER (&ZCMD, LIST, S, P, E, C, X)
```

to:

```
VER (&ZCMD, LIST, S, P, E, C, X, O)
```

7. Add this line to select the SDSF held output panel to the &ZSEL section of the panel definition:

```
O, 'PANEL (ZSDSFOP2) NEWAPPPL (ISF) OPT (H) '
```

8. Save and exit the file.
9. Invoke JOB/SCAN from the ISPF menu.
10. Verify that Standards Utility menu contains this line:

```
O Output - View held output using SDSF
```

## Increasing File Space Allocation

### To increase the space allocated for JLAB

1. From the ASG-JOB/SCAN Primary Menu panel, select the Product Administration Menu panel. (If the ASG-JOB/SCAN Primary Menu panel you see does not have this choice, consult the JOB/SCAN Product Administrator for assistance.)

- From the Product Administration panel, select Modify Options Members. The ISPF Edit member selection list for the SJJSPARM library appears.
- Select the SJJSPARM library member containing the File Allocations control statement (usually JSOPT02), and locate the File Allocations control statement for JLJAB.

#### Original JLJAB File Allocations Control Statement

```

EDIT ---- #COM2.SJJSPARM(JSOPT02) ----- COLUMNS 001 072
COMMAND ==>                                SCROLL ==> CSR
000009 *                                     **
000010 FILE ALLOCATIONS
000011 *
      .
      .
000040      JLJAB  =UNIT=SYSDA, PRIM=10, SEC=5, TRKS, NEW, DELETE
      .
      .
000043 REPORT OPTIONS
000044      OPTIONS IN EFFECT      =PDSOUT, WIDTH=132

```

- Edit the File Allocations control statement for JLJAB, increasing the space parameters.

#### JLJAB File Allocations Control Statement with Increased Space Requirements

```

EDIT ---- #COM2.SJJSPARM(JSOPT02) ----- COLUMNS 001 072
COMMAND ==>                                SCROLL ==> CSR
000009 *                                     **
000010 FILE ALLOCATIONS
000011 *
      .
      .
000040      JLJAB  =UNIT=SYSDA, PRIM=50, SEC=100, TRKS, NEW, DELETE
      .
      .
000043 REPORT OPTIONS
000044      OPTIONS IN EFFECT      =PDSOUT, WIDTH=132

```

- After saving the changes and exiting ISPF Edit, exit and reenter JOB/SCAN to initialize the JOB/SCAN session with the new SJJSPARM option values.

## Adding Site-Defined Functions

The User Function Detail Statement controls access to functions that are not part of the product, but which might need to be made available from the ASG-JOB/SCAN Primary Menu panel. The User Defined Function statements reside in JSOPT01. The Administrator can add up to two options to the ASG-JOB/SCAN Primary Menu panel, giving users access to other site applications, such as EDIT functions or utility programs. User Functions are available to every user; there are no group or USERID parameters to limit access.



Memory requirements of added user functions might require increased TSO region size.

### Example

JOB/SCAN builds the panel entry from the SELECT, NAME, and DESCRIPTION fields. JOB/SCAN inserts a dash (-) between NAME and DESCRIPTION. For example (see the figure for the effect of this statement):

---

```

ACCESS CONTROL
...
USER FUNCTION=PANEL=ISRUTIL,
  SELECT=UT,
  NEWAPPL=ISR,
  NAME=UTILITIES,
  DESCRIP='PERFORM UTILITY FUNCTIONS'

```

---

### The effect of a User Function Detail Statement in JSOPT01

```

----- ASG-JOB/SCAN PRIMARY MENU -----
SELECTION ===>

  0 - Hardcopy Output and JOB Statement Definition
  1 - Validate JCL
  2 - Change / Reformat JCL
  3 - Processing Options
  4 - Selection List Maintenance
  5 - Standards Utility Menu
  6 - Assign Alternate Options Members
  7 - Product Administration Menu
  8 - JOB/SCAN Usage Report
UT UTILITIES - PERFORM UTILITY FUNCTIONS
X - Exit

USERID - SYSUSR
PREFIX - SYSUSR
DATE - nnnn/01/12
TIME - 07:52
APPLID - JS

```

## Managing Database Interfaces

### DB2 Interface

During its run, JOB/SCAN:

- Parses and validates syntax of the DSN command and parameters.
- Verifies the existence of the DB2 program identified in the DSN RUN command.
- Verifies the existence of the DB2 library name specified in the DSN command.
- Verifies that the DB2 subsystem specified in the DSN command is an active MVS subsystem.

JOB/SCAN does not validate DB2 utility commands.

ASG provides a DB2 interface for extracting additional information for JOB/SCAN to validate (including DB2 syntax, dataset references, etc.). The information is available for batch DB2 jobs executing the DSN RUN command through the IKJEFT01, IKJEFT1A, and IKJEFT1B TSO driver programs.

See [Activating the IMS/DB2 Interface](#) for information about jobs accessing DB2 databases through IMS.

### Establishing the Interface

#### *To establish the DB2 interface*

1. Turn On the DB2 Interface

Turn on the interface by adding the DB2 parameter to the Program Product Interface (PPI) Run Option in the appropriate SJJSPARM library member. For example:

```
PROGRAM PRODUCT INTERFACE =DB2
```

2. Enable Control Card Access

Use the Access Control Card Run Option (ACC):

```
ACCESS CONTROL CARDS =YES
```

The ACCESS CONTROL CARDS statement tells JOB/SCAN to search PDS libraries for the DB2 control cards. In the case of IKJEFT01, IKJEFT1A, and IKJEFT1B, JOB/SCAN looks for DD SYSTSIN. If this DD refers to a PDS and member, JOB/SCAN reads that member and searches for DB2 control statements.

### 3. Code the Control Table Entries

Make sure the Control Table entries are coded in the appropriate SJJSPARM library member. (See [Control Table AB: Control Card DD Name](#) for details on Control Table AB.) For example:

```
.....+.....1.....+.....2.....+.....3.....+.....4
      ABSYSTSIN      IKJEFT01
      ABSYSTSIN      IKJEFT1A
      ABSYSTSIN      IKJEFT1B
```

The DB2 interface will be invoked automatically each time JOB/SCAN encounters a job that accesses a DB2 database through the DSN RUN command.

## Sample Batch DB2 Job Stream

---

```
//STEP EXEC PGM=IKJEFT01
//      .      .
//      .      .
//SYSIN DD *
      DSN  SYSTEM(DB2P)
      RUN  PROGRAM(PAYDB)  LIB (' PROD.SJJSLMOD)  PLAN (PAYDB)
      END
```

---

## IMS Interface

### Activating the IMS Interface

ASG provides an IMS interface for extracting additional information for JOB/SCAN to validate. The information is available for batch IMS jobs executing DFSRRCO0 programs through the DLI, BMP, and DBB functions.

See [Activating the IMS/DB2 Interface](#) for information on jobs accessing DB2 databases via IMS.JOB/SCAN Validation.

During its run, JOB/SCAN validates that:

- The IMS application program identified in the PARM parameter of the EXEC statement exists.
- The PSB specified in the Parm field and its associated DBDs can be found in the libraries specified by ddname IMS for DLI-type processing or ddname IMSACB for DBB-type processing.
- For BMP processing, the ACB specified in the Parm field is found in the ACB library.

### Establishing the Interface

#### *To establish the interface*

##### 1. Include the File Allocation Statement

Define the IMS ACB library in the File Allocation section of the appropriate SJJSPARM library member (BMP processing only).

Example:

```
ACBLIB =DSN=PROD.IMS.ACBLIB,SHR
```



JOB/SCAN performs special validation for batch jobs that access IMS databases using the DFSRRC00 driver with the DLI, BMP, and DBB options and the DFHDRP driver that accepts keyword parameters in the Parm field.

## 2. Turn on the IMS Interface

Turn on the interface by adding the IMS parameter to the Program Product Interface Run Option in the appropriate SJJSPARM library member. This Run Option is located in the shipped version of JSOPT02 member in the SJJSPARM library.

```
PROGRAM PRODUCT INTERFACE =IMS
```

**Or**

```
PROGRAM PRODUCT INTERFACE =(IMS, imsid)
```

where *imsid* is a 1- to 4- character IMS subsystem ID.

## 3. Code Control Table Entries

Make sure the appropriate Control Table entries are coded in the SJJSPARM library members. See [Control Tables](#) for details about Control Tables AD and AP for IMS.

IMS example:

```

.....+.....1.....+.....2.....+.....3.....+.....4
      ADDFSRRC00      UR
      APDFSRRC00      2

```

The IMS interface is invoked automatically whenever JOB/SCAN encounters the IMS driver program accessing databases through the DLI, BMP, or DBB functions.

### Activating the IMS/DB2 Interface

ASG provides an IMS/DB2 interface for extracting additional information for JOB/SCAN to validate. The information is available for batch IMS jobs executing DB2 programs through the DDITV02 DD control statements.

During its run, JOB/SCAN validates that:

- The IMS application program identified in the PARM parameter of the EXEC statement exists.
- The PSB specified in the Parm field and its associated DBDs can be found in the libraries specified by ddname IMS for DLI-type processing or ddname IMSACB for DBB-type processing.
- For BMP processing, the ACB specified in the Parm field is found in the ACB library.

In addition to the normal IMS validation process, control statements identified by the DDITV02 DD (for IMS/DB2) are validated.

### To establish the interface

1. Include the File Allocation Statement



Define the IMS ACB library in the File Allocation section of the appropriate SJJSPARM library member (BMP processing only).

Example:

```
ACBLIB =DSN=PROD.IMS.ACBLIB,SHR
```



JOB/SCAN performs special validation for batch jobs that access IMS databases using the DFSRRC00 driver with the DLI, BMP, and DBB options and the DFHDRP driver that accepts keyword parameters in the PARM field.

## 2. Turn On the IMS and DB2 Interfaces

Turn on the interface by adding the IMS and DB2 parameters to the Program Product Interface Run Option in the appropriate SJJSPARM library member. This run option is located in the shipped version of the JSOPT02 member in the SJJSPARM library.

IMS example with access to DB2 databases:

```
PROGRAM PRODUCT INTERFACE =IMS,DB2
```

Or

```
PROGRAM PRODUCT INTERFACE =(IMS,imsid)
```

where imsid is a 1 to 4 character IMS subsystem ID.

## 3. Code the Control Table Entries

Make sure the appropriate Control Table entries are coded in the SJJSPARM library members. See [Control Tables](#) for details on Control Tables AD and AP for IMS, and Control Table AB for IMS access of DB2 databases.

IMS example:

```
....+....1....+....2....+....3....+....4
  ABDDITV02      DFSRRC00
  ADDFSRRC00     UR
  APDFSRRC00     2
```

The IMS interface is invoked automatically whenever JOB/SCAN encounters the IMS driver program accessing databases through the DLI, BMP, or DBB functions.

# Managing Scheduler Interfaces

JOB/SCAN provides features for interfacing to a number of scheduling packages. This facilitates the rapid and direct tailoring of its operation to different and changing environments. See [Scheduler Interfaces](#) for more details about these features.

## Packages Supported

JOB/SCAN provides features that support the following scheduling packages:

### CA-7 Scheduler

JOB/SCAN provides sample programs and JCL to facilitate interfacing with the CA-7 Scheduler:

- Programs CA7J1 and CA7J2 are preprocessing interfaces that resolve any CA-7 #J statements in the input stream. CA7J1 will process a single schedule ID. CA7J2 is an updated version of the program, that will process

up to five schedule IDs.

See [Preprocessing CA-7 #J Directives](#). See [Advanced Topics for JCL Validation](#) and [Scheduler Interfaces](#) for additional information.

- CA7SCHED is a batch job interface to the CA-7 Scheduler for validating scheduled production jobs in run-time order. CA7SCHED is a three-step job that:
  1. Executes the CA-7 SASSBSTR program to produce a list of jobs in run-time order.
  2. Executes the ASG CA7INTF program to format the job names into a list of jobs to be validated by JOB/SCAN.
  3. Executes JOB/SCAN to validate the jobs in run-time order.

## CA Scheduler

The JOB/SCAN CA Scheduler interface CASCHED job:

- Executes the ASG program CASINTF to create a list of jobs to be validated by JOB/SCAN from a CA Scheduler forecasting report.
- Executes JOB/SCAN to validate the jobs in run-time order.



See [Scheduler Interfaces](#) for more details.

## CONTROL-M Scheduler

Two options are available for interfacing JOB/SCAN with the CONTROL-M Scheduler:

- A dynamic CONTROL-M interface that invokes special CONTROL-M routines to resolve %% variables in the JCL. This interface can be used for both foreground and batch JOB/SCAN execution.
- A batch interface to CONTROL-M for validating all jobs on the schedule for a given time period in run-time order. This interface extracts JCL from the CONTROL-M environment and writes it to a sequential file for JOB/SCAN to process.

## DCMS Scheduler

The JOB/SCAN DCMS Scheduler interface job:

- Executes the ASG program DCMSINTF to create a list of jobs extracted from the DCMS Scheduler.
- Executes JOB/SCAN to validate jobs in run-time order.

## ESP

The interface with the ESP Scheduler is fully described in the [User Information - JOB/SCAN with ESP](#) documentation.

## JOBTRAC

The JOBTRAC interface is invoked as a selection exit for JOB/SCAN. ASG provides a sample LE compliant interface program in SJJSSAMP called JTRC2XIT. The JOBTRAC interface simulates these JOBTRAC functions:

- Automatic JCL Override.
- JCL Management Facility.
- Automatic Date Insertion.

## TWS for z/OS

The interface with IBM's TWS for z/OS Scheduler is fully described in the *User Information - JOB/SCAN with TWS* documentation.

## Zeke

ASG provides two routines to interface with the Zeke scheduling package:

- ZEKEVARS is a Zeke interface invoked as a JOB/SCAN selection exit to provide the following functions:
  - Substitute appropriate values for Zeke symbolic parameters in the JCL.
  - Include the original JCL statements containing Zeke symbolic parameters, converted to comment statements, in the JCL.
  - Ensure that all JCL elements (for example, dataset names) are fully resolved.
- ZEKE14E is an interface that allows JOB/SCAN to be invoked as a Zeke user exit to provide the following functions:
  - Provide complete JCL Validation.
  - Ensure compliance with site standards.
  - Make successful JOB/SCAN Validation a prerequisite to Zeke job submission.

## Preprocessing CA-7 #J Directives

ASG provides two programs to preprocess JCL containing #J statements for the CA-7 Scheduler:

- CA7J1 can process a single schedule ID.
- CA7J2 can process from one to five schedule IDs.

The #J statements must be preprocessed before running JOB/SCAN - JCL validation. Except for the number of schedule IDs handled, the programs perform the same function.

## What the Preprocessor Does

When the preprocessor program (either CA7J1 or CA7J2) encounters a #JI or #JO statement, it compares the schedule ID and beginning and ending dates and times of that statement to values specified in the input parameter. See [Specifying the Parameter](#) Based on this comparison, the program either includes (#JI) or omits (#JO) the subsequent statements, up to the next #JI, #JO, or #JEND statement.

The program adds the prefix `//*` to each omitted statement and each #JI, #JO, and #JEND statement, so that they will be treated as comments and not be flagged as errors by JCL validation.

## Specifying the Parameter

Specify the parameter in the form: `iiyydddhhmmyydddhhmm` where:

`iii` is the schedule ID.

`yy` is the year.

`ddd` is the Julian day of the year.

hh is the hour (from 24-hour clock).

mm is the minute.

First date and time in the parameter are the beginning date and time; second date and time are the ending date and time.

For the selection to be in effect for one day only, omit the ending date from the parameter. The parameter shown in the below figure, Parm field entry, contains two schedule IDs with beginning dates only. The parameter shown in the below figure contains two schedule IDs with beginning dates and times, and ending dates and times.

The values in the input parameter are positional. If the #JI and #JO cards do not contain date or time fields, leave the date and time fields blank in the parameter. When using time without date, type zeros in the date field.

## Invoking the Preprocessor

### To invoke the preprocessor (either CA7J1 or CA7J2) in panel mode

- Type the program name in the JOB/SCAN - JCL Validation panel's Selection Exit field, and the preprocessor parameter in the corresponding Parm field. This example includes JCL between schedule IDs 10294332 and 12694357.

#### JOB/SCAN - JCL Validation Panel Specifying CA-7 #J Card Preprocessing

```

----- JOB/SCAN - JCL VALIDATION -----
COMMAND ==>

      F - Foreground execution mode          P - Processing Options
      B - Batch execution mode              E - Edit JCL member
                                          BR - Browse JCL member

Input Library  ==> SYSUSR.JCL
Input Type    ==> J                      (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ==> PDS                     (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> CA7TEST                 (Single member if BATCH with TRANSMIT)
Exclude Members ==>                       (Not valid for BATCH with TRANSMIT)
First Proclib ==>
Proclib Type  ==>                         (PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
Report Heading ==>
Display        ==> Y (Yes/No/ErrorsOnly)  Browse/View ==> B (B/V)
Hardcopy       ==> N (Yes/No/ErrorsOnly/SpaceOnly/JobErrors)
Options in Effect Listing ==> N (Y/N)      SJL PDSOUT ==> N (Y/N)

Selection Exit ==> CA7J2                  Parm ==> 10294332,12694357

Standards Name ==>                       Parm ==>

```

### To invoke the preprocessor from a batch job

- Include in the JLRCL ddname the Input Selection control statements.

#### Input Selection Control Statements Specifying CA-7 #J Card Preprocessing

```

INPUT SELECTION
SELECTION EXIT=CA7J2, '102943320830943321730,126943570830943571730'

```

## Managing Security Interfaces

JOB/SCAN provides a security interface invoked by the Security Exit Run Option. If the Security Exit Run Option specifies a security exit program name, JOB/SCAN calls the program to:

- prevent security violations by JOB/SCAN when it is validating an input job stream. Before opening a file, JOB/SCAN calls the security exit program to verify that the userid associated with the JOB/SCAN run has the proper access authority. If the JOB/SCAN userid does not have the proper access authority, JOB/SCAN writes an error message in the Structured JCL Listing and does not attempt to access the file.
- detect potential run-time security violations by *JCL in the input job stream* when it is submitted for execution. If the Security Exit Run Option specifies the DSNAME parameter, as well as a security exit program name, JOB/SCAN calls the security exit program to verify proper access authority to all dataset names specified in DD statements in the job stream being validated.
- pass it any USER or GROUP parameters found in the JOB statement or *//\*LOGONID* statement in the job being scanned. If the USER or GROUP specified in the job being scanned differs from the USER or GROUP running the JOB/SCAN job, then the security exit program must be running APF-authorized in order to perform the dataset name validation. If JOB/SCAN does not pass any USER or GROUP parameter to the security exit program, then the security exit program performs dataset name validation for the user running the JOB/SCAN job.
- validate authority for the expected type of access (read, update, or alter) for each dataset name it finds on a DD statement in the job being scanned. The expected access is determined by the DISP parameter on the DD statement, Control Table AC in the SJJSPARM library, an I/O code in the Short Description Run Option DEFAULT parameter, an I/O code in the Short Description file, or an I/O code in the comment field of either the DD statement being processed or a comment statement preceding the DD statement being processed.
- specify the USERID and GROUPID parameters in the Security Exit Run Option for the user who ultimately submits the job, if the job contains no USER or GROUP parameters in the JOB statement, but the job runs under a USERID that is different from the user running the JOB/SCAN job.

JOB/SCAN also calls the security exit program at termination to allow it to perform any necessary housekeeping, such as freeing allocated storage.

### Security Exit Run Option Control Statement

The JOB/SCAN security interface is activated by the Security Exit Run Option. The format of the Security Exit Run Option control statement is:

```
SECURITY EXIT=prgname,dsnameparm
```

where:

*prgname* is the program name of a security exit module that provides the interface between JOB/SCAN and security packages such as CA-ACF2, CA-TOP SECRET, and RACF. There is no default if this parameter is not coded.



*dsnameparm* is either *DSNAME*, if the security exit program is to perform dataset name validation, or *NODSNAME*, if the security exit program is not to perform dataset name validation. The default is *NODSNAME*. If *DSNAME* is coded without *prgname* also being coded, it has no effect.



Dataset name checking (`DSNAME` parameter) might require APF-authorization. See [Running the Product in an APF-authorized Environment](#) for information about running the APF-authorized.

### Security Exit Programs

ASG provides several security exit programs to use with JOB/SCAN. In addition to the load modules in the load library, the source code for these programs is included in the SJJSSAMP library. Installations with special security exit requirements can modify the programs as needed. This table lists security exit programs provided by ASG.

Program	Description
ACF2EXIT	Security exit program for interfacing with the CA-ACF2/MVS security package (provided for compatibility with security interfaces established for JOB/SCAN releases prior to 5.4). Does not support the dataset name validation function <code>DSNAME</code> (see the SAFx SJJSSAMP members for <code>DSNAME</code> support).
RACFEXIT	Security exit program for interfacing with the RACF and CA-TOP SECRET security packages (provided for compatibility with security interfaces established for JOB/SCAN releases prior to 5.4). Does not support the dataset name validation function <code>DSNAME</code> (see the SAFx SJJSSAMP members for <code>DSNAME</code> support).
SAF0	Security exit program that uses the System Authorization Facility (SAF) to interface with RACF, CA-ACF2/MVS, and CA-TOP SECRET. <div data-bbox="399 1073 1489 1234" style="border: 1px solid purple; padding: 5px; margin-top: 10px;"> <p> This program should not be used when running APF-authorized under ISPF with <code>DSNAME</code> coded in the Security Exit Run Option. It issues a <code>RACINIT</code> macro, and there is no way to specify a subpool for the <code>ACEE</code>.</p> </div>
SAF1	Security exit program that uses the third-party <code>RACHECK</code> instead of the <code>RACINIT/RACHECK</code> used by <code>SAF0</code> . <div data-bbox="399 1360 1489 1598" style="border: 1px solid purple; padding: 5px; margin-top: 10px;"> <p> This program works only with RACF Release 1.8 or greater, or the equivalent CA-ACF2/MVS or CA-TOP SECRET releases. <code>SAF1</code> uses subpool 251 for storage when invoked in an APF-authorized environment under ISPF. (This is because subpool 0 storage is deleted after the module is executed, while subpool 251 storage is not deleted until the job step task terminates.)</p> </div>
SAF2	Security exit program that performs the same as <code>SAF1</code> , except that it uses the <code>STATUS=ACCESS</code> parameter of the <code>RACROUTE</code> macro to eliminate logging of RACF violations (when the user doesn't have proper access authority). CA-ACF2/MVS (ACF2 6.1 gl 9510) and CA-TOP SECRET (TSS 4.4 gl 9506) also support the <code>STATUS=ACCESS</code> parameter. CA-ACF2

Program	Description
	requires APF authorization to use <code>STATUS=ACCESS</code> , unless a (CA-ACF2) SAFDEF record is defined with the NOAPFCHK keyword.

### Using the JOB/SCAN Security Interface

JOB/SCAN security interface provides a means to prevent security violations and S913 messages caused by unauthorized access attempts of a job. JOB/SCAN also validates all dataset access authorities using the security interface.

### Coding the Security Exit Run Option Control Statement

The Security Exit Run Option control statement activates the JOB/SCAN security interface. The following example shows a typical control statement entry for the SECURITY EXIT Run Option:

```
RUN OPTIONS
SECURITY EXIT    =SAF2, DSNNAME
```

In the example, `SAF2` is the program name of the security exit module that JOB/SCAN calls to perform access authority verification and dataset name validation, and `DSNAME` is the parameter necessary to activate dataset name validation.

#### To define a procedure for coding the Security Exit Run Option control statement

1. Choose a security exit program applicable for your site. Refer to [Managing Security Interfaces](#) for information about available security exit programs.
2. Code the Security Exit Run Option control statement in a suitable SJJSPARM library member. Refer to [Default Options Members, Relating Validation Panel Fields to SJJSPARM Options, and Modifying SJJSPARM Members](#). Ensure that this Security Exit Run Option control statement will not be overridden by another statement occurring later in the JOB/SCAN options member search order.
3. Code the `DSNAME` parameter on the Security Exit Run Option control statement if dataset name validation is required in the JCL Validation. Code the `NODSNNAME` parameter instead, if dataset name validation is not required.
4. Ensure that the Security Exit DSNAME Check field is set properly. When running JOB/SCAN JCL Validation from the JOB/SCAN - JCL Validation panel, the value entered in the Security Exit DSNAME Check field of the JOB/SCAN Processing Options panel overrides the parameter coded in the SJJSPARM library member. Access the JOB/SCAN Processing Options panel from either the ASG-JOB/SCAN Primary Menu panel or the JOB/SCAN - JCL Validation panel.

[JOB/SCAN Processing Options Panel](#) shows this field set to `Y`, which is equivalent to coding `DSNAME` in the Security Exit Run Option control statement. Setting this field to `N` is equivalent to coding `NODSNNAME` in the Security Exit Run Option control statement.



Changes to SJJSPARM library options will not take effect until JOB/SCAN has been initialized with the new values. After modifying SJJSPARM options, exit and reenter JOB/SCAN to initialize JOB/SCAN with the new values.

**JOB/SCAN Processing Options Panel**

```

----- JOB/SCAN PROCESSING OPTIONS -----
COMMAND ==>

Use Error Table           ==> N           (Y/N)
Condition Code Processing ==> N           (Y/N)
Catalog Access           ==> Y           (Y/N)
VTOC Access              ==> Y           (Y/N)
PDS Member Verification  ==> Y           (Y/N)
Security Exit DSNAME Check ==> Y       (Y/N)
Program Checking         ==> Y           (Y/N)
Disable SMS Processing   ==> N           (Y/N)
IDCAMS Processing        ==> Y           (Y/N)
Use SCHEDULER interface  ==> Y           (Y/N)
Job Limit                ==>           (1 to 999 or Blank For No Limit)
Access Control Cards     ==> Y           (Y/N)
Control Card Limit       ==>           (1 to 999 or Blank For No Limit)
Subsystem                ==> JES2       (JES2/JES3/None)
Suppress Error Messages  ==> NO        (A,W,E,0 to 99,or NO)
Extended Error Text      ==> S         (X=Extended,S=Short)
Press Enter to verify option values or <END> to leave and save option values.

```

5. Enter SHOW RUN to display the Summary of Options in Effect panel on the command line of the JOB/SCAN - JCL Validation panel to confirm that the Security Exit Run Option in effect on this panel is coded properly.

**Summary of Options in Effect Panel**

```

----- SUMMARY OF OPTIONS IN EFFECT ----- ROW 1 TO 50 OF 50
COMMAND ==>                                SCROLL ==> CSR

*****
RUN OPTIONS
*****
ACCESS CONTROL CARDS      =YES, LIMIT=000
      .
      .
      .
SECURITY EXIT             =SAF2 , DSNAME
      .
      .
      .
WRITE SELECTED JCL        =NO
***** BOTTOM OF DATA *****

```

**Access Authority Verification**

JOB/SCAN requires read access to files such as the user catalog, load libraries, procedure libraries, and control card libraries, and write access to output files.

The access authority verification function verifies that the JOB/SCAN batch job or TSO userid running JCL Validation has the necessary access authority for a file before actually attempting to access the file.

Before opening a file, JOB/SCAN calls the security exit program to verify that the userid running JOB/SCAN has proper access authority. If the userid does not have proper access authority, JOB/SCAN issues a message to the ISPF screen or the Structured JCL Listing, indicating that access has been denied to that file, and does not attempt to actually access the file; this prevents security violations and possible S913 messages.

Any of the security exit programs provided by ASG perform this function when activated by the Security Exit Run Option.



## Dataset Name Validation

This function allows JOB/SCAN to detect access authority violations in addition to other JCL errors when validating a job stream's JCL. JOB/SCAN can validate access authority either for the userid running the JOB/SCAN JCL Validation job or for different users (when running APF-authorized) identified in the job stream by USER and GROUP parameters in JOB statements or `//*LOGONID` statements.

If the security exit program rejects a userid, JOB/SCAN writes the following message to the Structured JCL Listing and continues the JCL Validation job without dataset name validation:

```
*ADVISORY - DSS4759A DSNAME SECURITY VALIDATION SUSPENDED FOR THIS JOB
```

If the security exit program finds that the userid for the JCL being validated does not have the proper access authority for the intended access to a dataset name on a DD statement in the input job stream, JOB/SCAN issues the following message to the Structured JCL Listing and continues the JCL Validation job:

```
***ERROR - DSS4757E - type ACCESS TO DATA SET DENIED TO USER userid
```

Any security exit programs provided by ASG (except ACF2EXIT and RACFEXIT) perform this function when activated by the Security Exit Run Option with the DSNAME parameter coded.

SAF0 should not be used when running APF-authorized under ISPF. This restricts JOB/SCAN from performing dataset name validation for different users identified in the job stream when using SAF0 as the security exit program.

## Optimizing JOB/SCAN's Resource Requirements

JOB/SCAN's resource requirements are optimized as follows:

### Initializing with Alternate SJJSPARM Member

The Initialization detail statement is found in the JSOPT01 member in the SJJSPARM library. It designates the member to be used for initialization in place of the default JSOPT02.

See [Initialization Detail Statement](#) for syntax and details about usage. See [Using Access Control Detail Statements](#) for details about using multiple detail statements. For example,

```
INITIALIZATION=GROUPID=ASG, MEMBER=DOCASG02
```

### Specifying a First PROCLIB for JCL Validation

When validating JCL, it might be necessary to specify a PROC library to be scanned before other PROC libraries; for example, to test a development PROC that will replace an existing PROC in a production job. The JOB/SCAN - JCL Validation panel provides a First PROCLIB field and the associated PROCLIB Type field for this purpose.

For example, consider the production job GOFORIT, which resides in PDS ASG.PROD.CNTL. A new PROC062 member, for performing the functions of STEP01, needs to be validated in its target environment before being promoted to the production system. That is, the new PROC062 JCL must be validated with the JCL of any unchanged production members and installation defaults.

**ASG.PROD.CNTL(GOFORIT) - Job Stream to Be Scanned**

```
//GOFORIT JOB (ACCTG), 'DO IT TO IT', CLASS=A
//JCLLIB DD DSN=ASG.PROD.PROCLIB, DISP=SHR
//STEP01 EXEC PROC062
//STEP02 EXEC PROC067
```

Assume that the development version of the PROC062 PROC is the only member in PDS ASG.TEST.PROCLIB. The production versions of all PROCs required by the GOFORIT job, including the old version of PROC062, are members in PDS ASG.PROD.PROCLIB.



Comments on the EXEC statements of PROC062 members shown in the figures indicate PROC version numbers.

**ASG.PROD.PROCLIB(PROC062) - PROC062 in Production PROC Library**

```
//PROC062 PROC
//PSTEP1 EXEC PGM=GRPT023 PROC062 - VERSION 003Z
//DD1 DD DSN=ASG.DEPT03.GOVARS, DISP=SHR
//DD2 DD DSN=ASG.DEPT03.GORPT01, DISP=(MOD, PASS)
```

**ASG.TEST.PROCLIB(PROC062) - PROC062 in Test PROC Library**

```
//PROC062 PROC
//PSTEP1 EXEC PGM=GRPT024 PROC062 - VERSION 004A
//DD1 DD DSN=ASG.DEPT03.GOVARS, DISP=SHR
//DD15 DD DSN=ASG.DEPT03.GODATA03, DISP=SHR
//DD2 DD DSN=ASG.DEPT03.GORPT01, DISP=(MOD, PASS)
```

**To validate the entire GOFORIT job, including the new PROC062 member, fill in the fields on the JOB/SCAN - JCL Validation panel as indicated**

1. In the First PROCLIB field, type the name of the library to be scanned first because it contains the new PROC062 member (for example, 'ASG.TEST.PROCLIB').
2. In the PROCLIB Type field, type PDS for partitioned dataset.
3. Fill in the remaining fields as appropriate.
4. Type F on the command line; press Enter to execute JOB/SCAN - JCL Validation.

## JOB/SCAN - JCL Validation Panel

```

----- JOB/SCAN - JCL VALIDATION -----
COMMAND ==>

      F - Foreground execution mode      P - Processing Options
      B - Batch execution mode           E - Edit JCL member
                                      BR - Browse JCL member

Input Library  ==> ASG.PROD.CNTL
Input Type    ==> J      (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ==> PDS    (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> GOFORIT (Single member if BATCH with TRANSMIT)
Exclude Members ==>      (Not valid for BATCH with TRANSMIT)
First Proclib ==> 'ASG.TEST.PROCLIB'
Proclib Type  ==> PDS    (PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
Report Heading ==>
Display        ==> Y (Yes/No/ErrorsOnly)  Browse/View ==> B (B/V)
Hardcopy       ==> N (Yes/No/ErrorsOnly/SpaceOnly/JobErrors)
Options in Effect Listing ==> N (Y/N)      SJL PDSOUT ==> N (Y/N)

Selection Exit ==>          Parm ==>
Standards Name ==>          Parm ==>

```

As a result, JOB/SCAN displays the Structured JCL Listing. The comment on the EXEC statement of PROC062 confirms that the development version of PROC062 was used for JCL Validation.

## Structured JCL Listing

```

JOB/SCAN          STRUCTURED JCL LISTING          03/30/95  11:02:53
-----
//GOFORIT  JOB (ACCTG),'DO IT TO IT',CLASS=A
//PROCLIB  DD  DSN=ASG.PROD.PROCLIB,DISP=SHR
//STEP01   EXEC PROC062
P1//PROC062  PROC
P1//PSTEP1  EXEC  PGM=GRPT024          PROC062 - VERSION 004A
P1//DD1    DD  DSN=ASG.DEPT03.GOVARS,DISP=SHR
P1//DD15   DD  DSN=ASG.DEPT03.GODATA03,DISP=SHR
P1//DD2    DD  DSN=ASG.DEPT03.GORPT01,DISP=(MOD,PASS)
//STEP02   EXEC PROC067
P1//PROC067  PROC
P1//PSTEP2  EXEC  PGM=GSUM014          PROC067 - VERSION 001M
P1//DD1    DD  DSN=ASG.DEPT03.GORPT01,DISP=(MOD,PASS)
*** ERROR SUMMARY AND COUNTS ***
  0  ADVISORY LEVEL
  0  WARNING LEVEL
  0  ERROR LEVEL
  0  TOTAL ISSUED          0 SUPPRESSED.
    THE HIGHEST SEVERITY CODE ENCOUNTERED WAS    00.
...

```

## Searching for a PROC During Validation

JOB/SCAN imitates the operating system PROC search method. . The OS first scans datasets explicitly and dynamically allocated to PROC libraries by the job stream JCL (that is, JOBPARM statement, JCLLIB statement, etc.). If the PROC is not found, then MVS searches the installation default PROC library, typically defined by the PROC00 DD statement in SYS1.PROCLIB member JES2.

JOB/SCAN's PROC search imitates the MVS PROC search method:

First, if in panel mode, it first searches the dataset allocated to JJJPC; this dataset is entered in the First PROCLIB field on the Validation panel. See [Specifying a First PROCLIB for JCL Validation](#) for more information about using a First PROCLIB.

Second, if PROC is not found in JJJPC datasets or the is field not used, JOB/SCAN searches datasets allocated to ddname JJJPB. These libraries are explicitly and dynamically allocated by the input job stream JCL (that is, JOBPARM statement, JCLLIB statement, etc).

Third, if the PROC is not found in JJJPB datasets, JOB/SCAN searches the datasets specified by the File Allocation statements in the SJJSPARM library members and allocated to JJJPA; these datasets are normally the same datasets as the MVS installation default PROC library.

### **Minimizing Processing Time for JCL Validation**

JOB/SCAN performs many different validation functions when validating JCL. A number of these functions are optional and can be activated or suspended by modifying the SJJSPARM Run Options control statements or the JOB/SCAN Processing Options panel values.

Other SJJSPARM Run Options and associated SJJSPARM library Control Tables can either exclude or select specific ddnames, PROCs, or programs for JCL Validation. This permits processing large input job streams while actually validating only those elements where validation is needed.

### **Eliminating Validation Functions**

Use the Run Options values in this section to reduce processing time when the specific validation functions are not necessary. Remember that the JOB/SCAN Processing Options panel values override the SJJSPARM library Run Options control statements when running JOB/SCAN from the JOB/SCAN - JCL Validation panel.

### **Access Control Cards (ACC)**

```
ACCESS CONTROL CARDS= NO
```

Equivalent JOB/SCAN Processing Options Panel Entry:

```
ACCESS CONTROL CARDS====> N
```

### **Catalog Access (CAC)**

```
CATALOG ACCESS = NO,NOVTOC,NOPGM,NOPDSMEM
```

Equivalent JOB/SCAN Processing Options Panel Entries:

```
Catalog Access====> N
VTOC Access====> N
Program Checking====> N
PDS Member Verification====> N
```

### **Processing Specifications (PSS)**

```
PROCESSING SPECIFICATIONS= NOCONDCODE
```

Equivalent JOB/SCAN Processing Options Panel Entry:

```
Condition Code Processing====> N
```

### Security Exit DSNAME Check

SECURITY EXIT = *prgname*,NODSNAME

Equivalent Specify SJJSPARM Options Panel Entry:

Security Exit DSNAME Check===> N

### SMS

SMS = NOACS,NOCAS

Equivalent JOB/SCAN Processing Options Panel Entry:

Disable SMS Processing===> Y

### Eliminating Job Stream Elements

When validating all elements of the input job stream is not necessary, use the SJJSPARM library Run Options and associated Control Tables in this section to reduce processing time. The choice of an “exclude” versus a “select” Run Option depends on which is easier to implement. Parameter values that are bold in the examples are defaults. No fields on the JOB/SCAN Processing Options panel correspond to these SJJSPARM Run Options.

#### Use PROC Exclude Table (UPCX)

USE PROC EXCLUDE TABLE=YES/NO

Associated Control Table: AN

#### Use PROC Select Table (UPCS)

USE PROC SELECT TABLE=YES/NO

Associated Control Table: AK

#### Use Program Exclude Table (UPGX)

USE PROGRAM EXCLUDE TABLE=YES/NO

Associated Control Table: AF

#### Use Program Select Table (UPGS)

USE PROGRAM SELECT TABLE=YES/NO

Associated Control Table: AJ

## Ensuring Equivalent JCL Validation Results

JOB/SCAN runs in three different modes: from ISPF panels, from batch for large libraries, and from ISPF Edit sessions using Edit macros. Because each mode can have different starting conditions (different SJJSPARM library members, Reformat rules, Control tables and so on), validation results for a given member can be different from mode to mode.

### JOB/SCAN Edit Macros versus JCL Validation Panel

This section provides instructions to ensure equivalent JCL validation results from the JOB/SCAN Edit macros with those

from the JOB/SCAN JCL-Validation panel.

### Option and Control Differences

This table shows the differences in operation between JCL Validation run from the JOB/SCAN Edit macros versus JCL Validation run from the JOB/SCAN - JCL Validation panel. Sources of options and controls shown the first row in the table override sources shown in the second row.

JOB/SCAN EDIT Macros	JOB/SCAN - JCL Validation Panels
SJJSPARM library <sup>a</sup> member JSOPTEM	JOB/SCAN - JCL Validation panel and JOB/SCAN Processing Options panel
	SJJSPARM library member JSOPT03
	SJJSPARM librarymember JSOPT02
	SJJSPARM library member JSOPT01

### Ensuring Equivalency

The best way to ensure equivalency of results obtained from the JOB/SCAN Edit macros and the JOB/SCAN - JCL Validation panel, is to make the options and controls established in the SJJSPARM library member JSOPTEM consistent with those entered into the JOB/SCAN Processing Options panel and the JOB/SCAN - JCL Validation panel. For example, if a First PROCLIB entry is used for operation from the JOB/SCAN - JCL Validation panel, then for operation from the JOB/SCAN Edit macros, a File Allocations detail statement for JLJPC is needed in JSOPTEM to ensure that the same PROC library is scanned ahead of any others. Code the File Allocations detail statement for ddname JLJPC, and specify the dataset name (DSN= parameter) of the dataset to be used as the First PROCLIB.

### *To verify the options and controls set by the JOB/SCAN - JCL Validation panel and the JOB/SCAN Processing Options panel*

1. Run the JCL Validation job from panel-assisted batch.
2. Browse the dataset specified by the DSN parameter of the File Allocations control statement for ddname JLINTRDR in the appropriate SJJSPARM library options member (usually JSOPT02).

The options and controls generated from the JOB/SCAN - JCL Validation panel and the JOB/SCAN Processing Options panel are shown as SYSIN data, except for First PROCLIB entries which appear as //JLJPC DD statements.

The DD statement for STEP01.JLJPC in figure shows the effect of entering a First PROCLIB field value of 'ASG.COM2TEST.PROCLIB' in the JOB/SCAN - JCL Validation panel.

Settings for other fields in the JOB/SCAN - JCL Validation panel and the JOB/SCAN Processing Options panel produce the control statements in the SYSIN data.

<sup>a</sup>Default SJJSPARM library members are shown here. Alternate SJJSPARM library members can be created for special circumstances. See [Alternate SJJSPARM Members](#) and [Generating Default JLJPA Specifications](#).

**JLINTRDR File**

```

BROWSE      ASGCOM2.WORK.INTRDR                Line 00000000 Col 001 080
Command ==>                                     Scroll ==> CSR
***** Top of Data *****
//ASGCOM2A JOB (ASGCS,132900,ASG,37), 'COM2TESTER',
//          NOTIFY=ASGCOM2,MSGCLASS=X
//*
/*ROUTE PRINT U609
//JOBPROC DD DSN=ASG.#COM2.PROCLIB,DISP=SHR
//STEP01 EXEC JS6PROC,OPT01=JSOPT01,OPT02=JSOPT02,PRM=S,
//          OPT0N=JSOPT03
//STEP01.JLJPC DD DISP=SHR,
//          DSN=ASG.COM2TEST.PROCLIB
//STEP01.SYSIN DD *
REPORT OPTIONS
    OPTIONS IN EFFECT=PRINT
    STRUCTURED JCL DISPLAY=ON
    STRUCTURED JCL LISTING=PRINT,ERRORTEXT=EXTENDED
RUN OPTIONS
    ACCESS CONTROL CARDS =YES,LIMIT=0
    CATALOG ACCESS =YES,VTOC,PDSMEM,PGM
    JCL CHANGE PARAMETERS=STRCHG80
    PREFIX =ASGCOM2
    PROCESSING SPECIFICATIONS =CONDCODE
    PROGRAM PRODUCT INTERFACE =IDCAMS
    REPORT HEADING = '
    SECURITY EXIT =*,DSN
    SUBSYSTEM =JES2
    SUPPRESS ERROR MESSAGES =
    USE ERROR TABLE =YES
    ...
INPUT SELECTION
    SELECT=GOJLJPA
    DSNAME=ASGCOM2.TEST.CNTL
    SOURCE TYPE=PDS

```

**Batch versus JCL Validation Panel**

This section describes how to obtain equivalent JCL Validation results from running in batch and from the JOB/SCAN - JCL Validation panel. The JOB/SCAN user must remember that options and controls affecting JOB/SCAN execution come from different sources for the two different modes of operation.

**Option and Control Differences**

This table shows the differences in operation between JCL Validation run in batch versus JCL Validation run from the JOB/SCAN - JCL Validation panel. Sources of options and controls shown the first row in the table override sources shown in the second row.

Batch Jobs <sup>a</sup>	JOB/SCAN - JCL Validation Panel
Batch job SYSIN dataset	JOB/SCAN - JCL Validation panel and JOB/SCAN Processing Options panel

<sup>a</sup>Although the batch job can allocate different files to ddname JLRCL instead of the same SJJSPARM library members used by JCL Validation run from the JOB/SCAN - JCL Validation panel, allocating the same SJJSPARM members will ensure equivalent results.

Batch Jobs<sup>a</sup>

## JOB/SCAN - JCL Validation Panel

SJJSPARM library member JSOPT03<sup>b</sup>

SJJSPARM library member JSOPT02

SJJSPARM library member JSOPT01

**Ensuring Equivalency**

The best way to ensure equivalent results obtained from a JCL Validation run from batch and from the JOB/SCAN - JCL Validation panel, is to be sure that the JCL of the job submitted in batch allocates to ddname JLRCL the same SJJSPARM library options members used by JCL Validation run from the JOB/SCAN - JCL Validation panel, plus an additional dataset (such as SYSIN) containing options and controls consistent with those set from the JOB/SCAN - JCL Validation panel and the JOB/SCAN Processing Options panel.

For example, if a First PROCLIB entry is required for operation from the JOB/SCAN - JCL Validation panel, then for operation in batch, a DD statement for ddname JLJPC is required to ensure that the same PROC library is scanned ahead of any others. Code the JLJPC DD statement similar to the one shown in the below figure, and specify the dataset name (DSN=) parameter of the dataset to be used as the First PROCLIB.

***To verify SJJSPARM library options members and options and controls set by the JOB/SCAN - JCL Validation panel and the JOB/SCAN Processing Options panel***

1. Run the JCL Validation job from panel-assisted batch
2. Browse the dataset specified by the DSN parameter of the File Allocations control statement for ddname JLINTRDR in the appropriate SJJSPARM library options member (usually JSOPT02).

The SJJSPARM library options members used are named in parameters of the EXEC statement, and the options and controls generated from the JOB/SCAN - JCL Validation panel and the JOB/SCAN Processing Options panel are shown as SYSIN data, except for First PROCLIB entries which appear as //JLPC DD statements.

In the below figure, the values for EXEC statement parameters OPT01, OPT02, and OPT0N, are names of the SJJSPARM library options members used for the JCL Validation job. In this example, the member names are the names of the default members, JSOPT01, JSOPT02, and JSOPT03. The DD statement for STEP01.JLJPC shows the effect of entering 'ASG.COM2TEST.PROCLIB' in the First PROCLIB field of the JOB/SCAN - JCL Validation panel.

Settings for other fields in the JOB/SCAN - JCL Validation panel and the JOB/SCAN Processing Options panel produce the control statements in the SYSIN data.

<sup>a</sup>Although the batch job can allocate different files to ddname JLRCL instead of the same SJJSPARM library members used by JCL Validation run from the JOB/SCAN - JCL Validation panel, allocating the same SJJSPARM members will ensure equivalent results.

<sup>b</sup>Default SJJSPARM library members are shown here. Alternate SJJSPARM library members can be created for special circumstances. See [Alternate SJJSPARM Members](#) and [Generating Default JLJPA Specifications](#).



**JLINTRDR File**

```
//ASGCOM2A JOB (ASGCS,132900,ASG,37),'COM2TESTER',
//          NOTIFY=ASGCOM2,MSGCLASS=X
//*
/*ROUTE PRINT U609
//JOBPROC DD DSN=ASG.#COM2.PROCLIB,DISP=SHR
//STEP01 EXEC JS6PROC,OPT01=JSOPT01,OPT02=JSOPT02,PRM=S,
//          OPT0N=JSOPT03
//STEP01.JLJPC DD DISP=SHR,
//          DSN=ASG.COM2TEST.PROCLIB
//STEP01.SYSIN DD *
REPORT OPTIONS
  OPTIONS IN EFFECT=PRINT
  STRUCTURED JCL DISPLAY=ON
  STRUCTURED JCL LISTING=PRINT,ERRORTEXT=EXTENDED
RUN OPTIONS
  ACCESS CONTROL CARDS =YES,LIMIT=0
  CATALOG ACCESS =YES,VTOC,PDSMEM,PGM
  JCL CHANGE PARAMETERS=STRCHG80
  PREFIX =ASGCOM2
  PROCESSING SPECIFICATIONS =CONDCODE
  PROGRAM PRODUCT INTERFACE =IDCAMS
  ...
INPUT SELECTION
  SELECT=GOJLJPA
  DSNAME=ASGCOM2.TEST.CNTL
  SOURCE TYPE=PDS
```

**Using DD Statements Instead of the DSN Statement**

When using a program to generate SELECT statements or when concatenating multiple input JCL libraries, specify DD statements instead of using the DSN detail-level statement to specify the libraries containing the JCL to be validated.

**Default ddname: JLJOB**

The default ddname is JLJOB. The statements may be placed in the JOB/SCAN execution JCL or within the JOB/SCAN PROC. The DSN detail-level statement can be omitted. For example,

```
//JLJOB DD DSN=MY.JCLLIB,DISP=SHR
//      DD DSN=PROD.CNTL,DISP=SHR
```

**Alternate Method: File Allocations**

An alternate method of Input Selection is to use the File Allocation section of the SJJSPARM library member. For example,

```
FILE ALLOCATIONS

      JLJOB          =DSN=MY.JCLLIB,SHR
                   =DSN=PROD.CNTL,SHR
```



If both the JLJOB ddname and DSN detail-level statements are included, JOB/SCAN uses the library specified in the DSN statement.

## DDNAME Detail-level Statement

### To use a ddname other than JIJOB

- Use the DDNAME detail-level statement in the Input Selection. For example,

```
//INJCL DD DSN=MY.JCLLIB,DISP=SHR
//      DD DSN=PROD.CNTL,DISP=SHR

...
//SYSIN DD *
INPUT SELECTION
        DDNAME=INJCL
        SELECT=JOB001
```

## Generating Default JLPA Specifications

JOB/SCAN provides a means of automatically identifying the default PROCLIBs searched by JOB/SCAN as it expands PROCs, and of generating default JLPA File Allocations statements. The function is performed by the COBOL standards program JSXTDJES, which is a member in the JOB/SCAN SJSSTDL dataset. JSXTDJES calls the assembler module J06YJES, which is in the JOB/SCAN SJJSLMOD load library. The JOB/SCAN Product Administrator can execute JSXTDJES when JOB/SCAN is installed, and any time the PROCLIB concatenations change in the JES2 Startup JCL. Executing JSXTDJES takes the place of manually specifying the JLPA File Allocations statements in SJJSPARM library member JSOPT02, as done in previous releases of JOB/SCAN.



JSXTDJES only generates specifications for default (PROCO0) PROCLIBs. JSXTDJES does not work with dynamically generated PROCLIBs.

## Running JSXTDJES in Batch

### To invoke JSXTDJES in batch

- Submit a job similar to the job shown in figure. The job must contain the Run Options control statement `STANDARDS NAME =JSXTDJES` and these Input Selection control statements:
  - `DSN =SYS1.PROCLIB`
  - `SOURCE TYPE =PROC` (This is always PROC.)
  - `SELECT =jesstartup` (`jesstartup` is usually JES2.)

### Example Batch Job for Default JLPA Specification Generation

```
//GENJLJPA JOB (ACCT), 'PGMR NAME'
//JOBPROC DD DSN=YOUR.PROCLIB
//STEP01 EXEC JS6PROC,PRM=S
//STEP01.SYSIN DD *
INPUT SELECTION
  DSN          =SYS1.PROCLIB
  SOURCE TYPE  =PDS
  SELECT       =JES2
RUN OPTIONS
  STANDARDS NAME =JSXTDJES

REPORT OPTIONS
  OPTIONS IN EFFECT      =PRINT
  STRUCTURED JCL LISTING =PRINT
```

## Running JSXTDJES from the JCL Validation Panel

### To invoke JSXTDJES from the JOB/SCAN - JCL Validation panel

1. Type 'SYS1.PROCLIB' in the Input Library field.
2. Type P (proc) in the Input Type field.
3. Type the name of the JES2 Startup JCL member in the Select Members field. (Usually this is JES2.)
4. Type JSXTDJES in the Standards Name field.
5. Type B (batch) or F (foreground) on the command line.
6. Press Enter.



JOB/SCAN normally inserts a First PROCLIB specification in the panel when the Input Type field is P.

### JOB/SCAN - JCL Validation Panel: Foreground Default JLPA Specification Generation

```

----- JOB/SCAN - JCL VALIDATION -----
COMMAND ==>

      F - Foreground execution mode          P - Processing Options
      B - Batch execution mode              E - Edit JCL member
                                          BR - Browse JCL member

Input Library  ==> 'SYS1.PROCLIB'
Input Type    ==> P          (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ==> PDS       (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> JES2
Exclude Members ==>
First Proclib ==>
Proclib Type  ==>          (PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
Report Heading ==>
Display        ==> Y (Yes/No/ErrorsOnly)  Browse/View ==> B (B/V)
Hardcopy       ==> N (Yes/No/ErrorsOnly/SpaceOnly/JobErrors)
Options in Effect Listing ==> N (Y/N)      SJL PDSOUT ==> N (Y/N)

Selection Exit ==>          Parm ==>
Standards Name ==> JSXTDJES  Parm ==>

```

### JSXTDJES Output: Options Member JSOPTJ2

JSXTDJES creates a new SJJSPARM library options member named JSOPTJ2, which contains File Allocations statements for JLPA. JOB/SCAN uses the File Allocations statements for JLPA in the SJJSPARM library options member JSOPTJ2 to define the default PROCLIB for JCL Validation instead of those in member JSOPT02, if the Run Options in effect include a Program Product Interface statement specifying JES2PROC and include this dd statement in the JLRL concatenation:

```
//          DD  DISP=SHR, DSN=&DJSNAME..SJJSPARM(JSOPTJ2)
```

**JSOPTJ2 SJSPARM Library Member Generated by JSXTDJES Standards Program**

```

** ** JSOPTJ2 ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **
FILE ALLOCATIONS
  JLJPA   =DSN=SYS1.PROCLIB, SHR
         =DSN=LCS1.CL.PROCLIB, SHR
         =DSN=LCS1.IPP.PROCLIB, SHR

```

The below figure shows Run Options control statements including a Program Product Interface statement specifying JES2PROC (to direct JOB/SCAN to use JSOPTJ2 JLJPA File Allocations statements instead of JSOPT02 JLJPA File Allocations statements).

**Program Product Interface Entry Using JLJPA File Allocations Statements from JSOPTJ2 Instead of JSOPT02**

```

RUN OPTIONS
  PROGRAM PRODUCT INTERFACE=IDCAMS, EZPROCLIB
  PROGRAM PRODUCT INTERFACE=JES2PROC
  CATALOG ACCESS      =YES, VTOC, PDSMEM
  ACCESS CONTROL CARDS =YES
  SECURITY EXIT        =SAF2, DSNNAME
  SMS                  =NOACS, NOCAS

```

## Managing JOB/SCAN's Output Messages

This topic describes how to manage JOB/SCAN's Output Message

### Modifying Extended Error Message Text

JOB/SCAN provides the option to include extended error messages in the Structured JCL Listing (or to use the JMSG Edit macro for JEM Edit macro results) when more information than the short messages provide is needed.

This product distributes extended error message text in these two members in the SJJSPARM library:

- DSSMSG00—Contains the extended error message text.
- DSSMSG90—Contains the extended error message text for user exits such as standards and security exits.

### SJJSPARM Member DSSMSG00

SJJSPARM library member DSSMSG00 contains extended error message text for the JOB/SCAN product error messages. These messages are in the same format and use the same rules as the extended error message text for user exit error messages in member DSSMSG90. Read the rules in SJJSPARM member DSSMSG90 for guidance in writing or modifying the messages in this member.

### SJJSPARM Member DSSMSG90

SJJSPARM library member DSSMSG90 contains dummy message DSS9000 as a template for creating additional extended error messages. It also contains several examples of extended error messages and detailed rules for editing the extended error message text.

#### *To add or modify extended error messages*

1. Edit DSSMSG90 either directly from the ISPF panels or from the Modify Options Members option of the JOB/SCAN Product Administration panel.

2. Follow the directions at the beginning of the member.

### SJJSPARM Library Member DSSMSG90 (Partial)

```

EDIT ---- JSnnn.SJJSPARM(DSSMSG90) ----- COLUMNS 001 072
COMMAND ==>                                SCROLL ==> CSR
***** ***** TOP OF DATA *****
000100 * What follows are examples of messages to assist you in entering your
000200 * own messages.
000300 *
000400 * Message DSS9000 is a template for your use.
000500 *
000600 * The other messages are examples which were extracted from the sample
000700 * JOB/SCAN STANDARDS User Exit CSMP and the sample Security Exit
000800 * SAF0.
000900 *
001000 * Rules for entering text:
001100 *   - the error message number, which begins with "DSS", must start in
001200 *     column 1, and "DSS" must be in caps,
001300 *   - the short message (which is currently optional here), if entered,
001400 *     should begin in column 11 on the same line as the error message
001500 *     number. The short message can be up to 52 characters in length.
001600 *   - the extended text uses columns 2 - 72, can use an unlimited
001700 *     number of lines, and is completely in free form.
001800 *   - comments are indicated by an asterisk (*) in column 1. They may
001900 *     appear anywhere within a message and they will not be included
002000 *     during the display of the error text.
002100 *   - while it is recommended that error message numbers be kept in
002200 *     sequence, it is not a requirement at this time.
002300 *   - it is recommended that the severity letter (A, W, or E) be
002400 *     omitted from the error message number. It is not used by the
002500 *     JOB/SCAN Extended Message handler.
002600 *
002700 *           <===== 52 characters =====>
002800 DSS9000   This is the short error message text up to 52 chars.
002900   xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

```

### Displaying Optional Extended Error Message Text

After using the JOB/SCAN Edit Macro (JEM), use the JMSG Edit Macro to display an extended version of any JCL Validation error message. These extended messages are helpful if further clarification is required.

### Using JMSG

#### *To view an extended error message*

1. Execute the JEM Edit macro. (See [JEM Edit Macro](#) for instructions.)
2. Type JMSG on the command line.
3. Position the cursor on the JEM label preceding an error or group of errors.  
JEM labels appear in the line number field on the left of the screen in the form.Jxxxx.
4. Press Enter.

If only one error is associated with the label, its extended text is displayed. However, if multiple errors are associated with the label, an Extended Message Selection panel appears.

5. In the case of multiple errors, type S next to each message for which extended text is desired, and press Enter.

Press End to move from one message to the next.

6. Press End after viewing the message or messages to return to the JCL.
7. Press End to terminate the ISPF Edit session.

Below figure shows the Extended Message Selection panel, which allows selection of messages:

#### Extended Message Selection Panel

```

----- JEM - Extended Message Selection -----RPW 1 TO 2 of 2
COMMAND ==>                                SCROLL ==> CSR

  Sel Message      Short Message Text
  Number

  s  DSS4050E      DATA SET NOT FOUND IN CATALOG
     DSS8900A      DSN = "ASG.PAYROLL.MASTER"
*****

```

When an error message is selected, the extended text is displayed on the Extended Message Text panel.

#### Message after Expansion

```

----- JEM - Extended Message Selection -----RPW 1 TO 9 of 9
COMMAND ==>                                SCROLL ==> CSR

DSS4050E      DATA SET NOT FOUND IN CATALOG

*****
*  DSS4050  Data set not found in catalog          *
*  Related IBM message - IEF283I                  *
*  The data set specified in the DD statement with a DISP of *
*  OLD or SHR could not be found in the catalog.  This could *
*  be caused by a misspelled DSN or due to the fact that the *
*  data set may have been uncataloged previously in the set *
*  of jobs being scanned.                          *
*****

```

### Assigning JMSG to a PF Key

*To facilitate using JMSG, a PF key can be set to the value of JMSG*

1. Position the cursor at the desired JEM LABEL.
2. Press the PF key to obtain the extended message text.

### Suppressing JCL Validation Error Messages

There might be occasions when JCL Validation error messages should be suppressed to accommodate the site-specific environment. JOB/SCAN provides the means of suppressing error messages in the following ways:

- By severity level with the Suppress Error Messages run option.
- For specific error codes, or combinations of specific error code and specific program name or ddname, with Control Table AW.
- According to user-defined criteria with the JOB/SCAN Standards Facility.

## Suppress Error Messages (SEM) Run Option

The Suppress Error Messages run option suppresses, by severity level, error messages that are normally displayed in the Structured JCL Listing. If the Suppress Error Messages run option is in effect, JOB/SCAN suppresses both JOB/SCAN and Standards Facility messages. The format of the Suppress Error Messages run option is:

```
SUPPRESS ERROR MESSAGES =NO/A/W/E/nn
```

The list in this table indicates the severity level of messages suppressed by each Parameter value:

Value	Messages Suppressed
NO	None (Default)
A	Advisory
W	Advisory and warning
E	Advisory, warning, and error
nn	All messages with a severity level equal to or less than nn. (nn = a decimal value from 0 to 99).

When JOB/SCAN is run from the JOB/SCAN - JCL Validation panel, the Suppress Error Messages field of the JOB/SCAN Processing Options panel determines the value in effect for the Suppress Error Messages run option. JOB/SCAN normally resets the value of the Suppress Error Messages field of the JOB/SCAN Processing Options panel to NO for each JOB/SCAN session.

## SEMRESET Run Option

The SEMRESET run option can cancel resetting the value of the Suppress Error Messages run option. The SEMRESET run option format is:

```
SEMRESET =YES/NO
```

The purpose of each parameter value:

Value	Purpose
YES (default)	Reset the value of the Suppress Error Messages field of the JOB/SCAN Processing Options panel to NO for each JOB/SCAN session.
NO	Save the value of the Suppress Error Messages field of the JOB/SCAN Processing Options panel for subsequent JOB/SCAN sessions.

## Control Table AW

Control Table AW provides the means of suppressing or changing the severity level of JCL error messages for specified JOB/SCAN error numbers. A Control Table AW entry can be coded to suppress any of the following error messages or

change their severity level:

- All occurrences of the specified error number.
- Only those occurrences in JCL for a specified program name.
- Only those occurrences for a specified ddname.
- Only those occurrences for a specified ddname in JCL for a specified program name.

See [Control Table AW: Suppress JCL Error Messages](#) for details about the AW control table.

### **Related Run Option: Use Error Table (UET)**

The Use Error Table run option must be set to `YES` for JOB/SCAN to use the entries in Control Table AW. (The default, when not coded, is `NO`.)

When running JOB/SCAN from the JOB/SCAN - JCL Validation panel, the value (Y for yes, or N for no) entered from the JOB/SCAN Processing Options panel determines the setting.

When running in batch, the value set by the last occurring Use Error Table run option in ddname JLRCL (usually concatenated SJJSPARM library members) determines the setting.

The Use Error Table Run Option format is:

```
USE ERROR TABLE YES/NO
```

### **Control Table AW Search Precedence**

JOB/SCAN gives precedence to the Control Table AW entry occurring *first* when there is a conflict. For example, if two entries in the table specify altering the severity level of all occurrences of the same error number to different values, then the entry occurring first prevails. If two entries in the table specify altering the severity level of the same error number to different values, but specify different programs or ddnames, then there is no conflict and JOB/SCAN will process and use both entries.

Entries that suppress error messages cannot conflict with each other. Because all error suppression entries do the same thing, but possibly in different instances, their effects can only be additive. That is, *all* of the Control Table AW error message suppression entries for a JOB/SCAN error number will take effect.

### **Example Error Message Suppression Entries**

The entries in Control Table AW suppress error messages as follows:

- The first entry, for JOB/SCAN error number DSS4270, suppresses all occurrences of error number DSS4270.
- The next two entries, for JOB/SCAN error number DSS5110, suppress DSS5110 messages only for ddnames IDCUT1 and IDCUT2 occurring in program IDCAMS.
- The next entry, for JOB/SCAN error number DSS8110, suppresses DSS8110 messages for any ddnames, but only in program PRGALT4.
- The next entry, for JOB/SCAN error number DSS8605, suppresses DSS8605 messages in any program, but only for ddname DD1.
- The last three entries, for JOB/SCAN error number DSS4500, suppress all DSS4500 messages. The order of the three entries does not affect the result because there is no conflict and the effects are additive.



**Control Table AW Error Message Suppression Entries**

```
CONTROL TABLES
  AWDSS4270      XX
  AWDSS5110E    XX IDCAMS  IDCUT1
  AWDSS5110E    XX IDCAMS  IDCUT2
  AWDSS8110     XX PRGALT4
  AWDSS8605     XX          DD1
  AWDSS4500     XX PRGALT7
  AWDSS4500     XX          DD2
  AWDSS4500     XX
```

**Standards Facility**

Except for DSS0xxx error messages, a JOB/SCAN standards program can change the severity of any JOB/SCAN messages, or suppress the messages altogether. See [Changing or Suppressing JOB/SCAN Error Messages](#) and [Control Table AW: Suppress JCL Error Messages](#) in documentation for more information about how standards programs suppress error messages.

# Options

This topic provides detail information about the administrative options groups.

- [System Identification Group](#)
- [Access Control Group](#)
- [File Allocations Group](#)
- [Run Options Group](#)
- [Report Option Group](#)
- [Control Table Group](#)
- [Input Selection Group](#)

## System Identification Group

The detail statements comprising the System Identification group reside only in SJJSPARM library member JSOPT01. Values specified in the detail-level statements are provided by ASG.

### Format

```
SYSTEM IDENTIFICATION
    TITLE1='your_title1_description'
```

See [Detail-Level Statements Format Rules](#) when entering detail statements for System Identification statements.

### Parameter

Parameter	Function
TITLE1	Maximum of 32 characters, enclosed in single quotations. Usually the company name or division. JOB/SCAN uses this parameter in all reports.

### Example

#### Modified System Identification Statement in JSOPT01

```
EDIT          #USR1.SJJSPARM(JSOPT01) - 01.99          Columns 00001 00072
Command ==>                                         Scroll ==> PAGE
***** ***** Top of Data *****
000001 ** JSOPT01  ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **^
000002 **                                                                                                     **
000003 **      THIS MEMBER ENABLES ACCESS TO JOB/SCAN AND                                               **
000004 **      CONTROLS USER ACCESS TO PANELS AND EXTENDED FUNCTIONS.                                     **
000005 **                                                                                                     **
000006 ** ** ** ** ** ** ** ** ** ** ** **^
000007 *                                                                                                     **
000008 SYSTEM IDENTIFICATION
000009     TITLE1='WIDGET MANUFACTURING INC'
000010 ACCESS CONTROL
000011     INITIALIZATION           =ALL
```

## Access Control Group

The detail statements comprising the Access Control group reside in SJJSPARM library member JSOPT01. These statements control user access to the ASG-JOB/SCAN Primary Menu selections and assigned options members used by these functions. If a user is denied access to a function, the entry for the function is not included on the user's ASG-JOB/SCAN Primary Menu panel.

Detail Statement	Function
ADMINISTRATOR	Designates access to the Product Administration panel.
INITIALIZATION	Designates member to be used for initialization in place of the default JSOPT02.
JOB/SCAN	Specifies user access to JOB/SCAN from the ASG-JOB/SCAN Primary Menu panel; grants access privilege to the Standards Facility.
JOB/SCAN USAGE REPORT	Designates user access to the Product Usage Reports from the ASG-JOB/SCAN Primary Menu panel.
JOB/SCAN USAGE REPORT FILE	Enables the writing of product usage data to the usage file.
OPTIONS	Designates user access to the Assign Alternate Options Members panel for specifying personal options members.
SELECT LISTS	Provides user access to the Selection List Maintenance Facility panel for creating lists of jobs to be validated or documented as a group.
STANDARDS PROMPTER	Controls which users can access the Standards Utility Menu panel from the ASG-JOB/SCAN Primary Menu panel.
USER FUNCTION	Designates user access to non-product functions from the ASG-JOB/SCAN Primary Menu panel.

### Format

ACCESS CONTROL  
DETAIL-LEVEL STATEMENTS

See [Detail-Level Statements Format Rules](#) when entering detail statements for Access Control statements.

### Parameters

In general, the Access Control detail statements use these parameters:

Parameter	Function
USERID	Specifies a user, list of users, or mask.
GROUPID	Specifies a group, list of groups, or mask.
ALL	Specifies all users.
NONE	Specifies no users.
MEMBER	Assigns an alternate SJJSPARM library member to a user or group.

These examples show how to use detail statement parameters:

USERID=MYID	One user: type the user name.
USERID= (MYID, YOURID)	More than one user: separate the user names with commas and enclose in parentheses.
USERID=COM*****	A group of users: use a USERID mask.
GROUPID=GL012	One group: type the group name.
GROUPID= (GL012, GL057)	More than one group: separate the group names with commas and enclose in parentheses.
GROUPID=GL*****	To specify a collection of groups, use a GROUPID mask.

### Using Access Control Detail Statements

Each Access detail statement grants or denies some users access to a particular JOB/SCAN function. Code as many detail statements as needed to cover all the users, keeping in mind the order of the detail statements is important: JOB/SCAN uses the *first* statement that matches the current userid or groupid.

Grant user access according to site requirements by writing detail statements in this order:

- Specific userids, then specific groupids.
- Userids and groupids with masks.
- ALL or NONE.

For example:

```
ACCESS CONTROL
  ADMINISTRATOR=USERID=ASGCOM1
  ADMINISTRATOR=GROUPID=ASGGRP1
  ADMINISTRATOR=NONE
```

In this example, ASGCOM1 is a specific userid and ASGGRP1 is a system-defined group which get Administrator privileges. No other userids and groups (because they were not specified in other USERID or GROUPID statements) get Administrator privileges.



By default, the MVS ACEE control block determines the group a user is associated with.

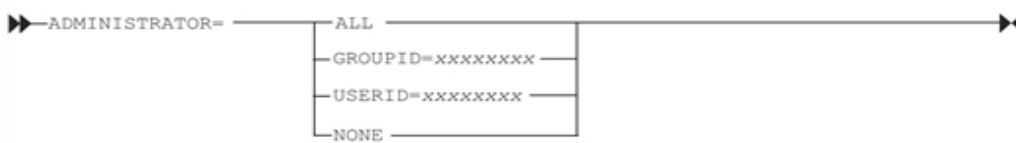
### Administrator Detail Statement

The Administrator detail statement controls which users can access the Product Administration panel from the ASG-JOB/SCAN Primary Menu panel.



To preserve control over SJJSPARM library members, ASG recommends that access to the Product Administration panel be limited to a small number of users; do not use ALL in this detail statement.

### Format



### Example

```
ADMINISTRATOR=USERID=ASGCOM1
```

### Parameters

See [Parameters](#) for additional information about entering userids and groupids.

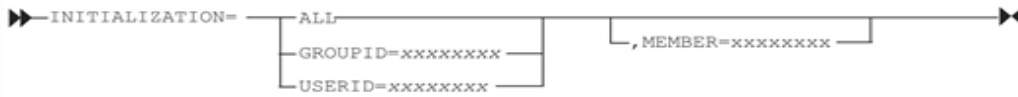
Parameter	Function
ALL	Grants all users access to the Product Administration panel from the ASG-JOB/SCAN Primary Menu panel. ASG does not recommend the use of this parameter.
NONE	Denies all users access to the Product Administration panel from the ASG-JOB/SCAN Primary Menu panel. ASG does not recommend the use of this parameter.
GROUPID=xxxxxxx	Grants specific groups of users access to the Product Administration panel from the ASG-JOB/SCAN Primary Menu panel.
USERID=xxxxxxx	Grants specific users, multiple users, or groups of users access to the Product Administration panel from the ASG-JOB/SCAN Primary Menu panel.

### Initialization Detail Statement

The Initialization detail statement assigns the member to be used for initialization in place of the default JSOPT02 for

specified users or groups.

**Format**



**Example**

INITIALIZATION=GROUPID=ASG, MEMBER=JSNEW02

**Parameters**

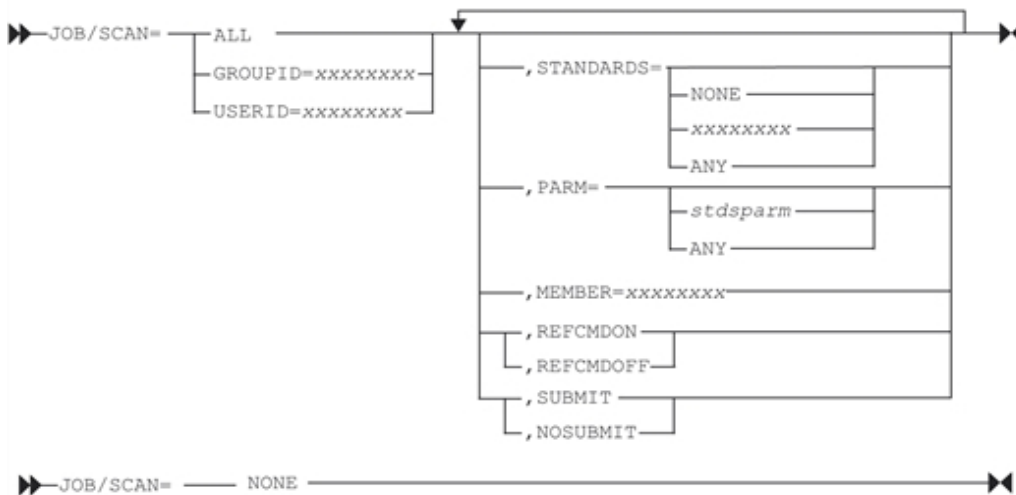
See [Parameters](#) for additional information about entering userids and groupids.

Parameter	Function
ALL	Assigns to all users a particular member for initialization, as specified in the MEMBER= parameter. If no alternate member is assigned, ALL is the only parameter allowed on this statement.
GROUPID=xxxxxxxx	Assigns to a specific group or multiple groups of users a particular member for initialization, as specified in the MEMBER= parameter.
USERID=xxxxxxxx	Assigns to a specific user or multiple users, a particular member for initialization, as specified in the MEMBER= parameter.
MEMBER=xxxxxxxx	Designates an assigned options member to override JSOPT02.

**JOB/SCAN Detail Statement**

This detail statement controls user access to JOB/SCAN through the ASG-JOB/SCAN Primary Menu panel. This statement grants access to the Standards Facility and REFORMATTER and assigns an alternate option member to override JSOPT03.

**Format**



### Example

JOB/SCAN=ALL, STANDARDS=ANY, PARM=, SUBMIT

### Parameters

See [Parameters](#) for additional information about entering userids and groupids.

Parameter	Function
ALL	Grants all users access to JOB/SCAN.
NONE	Denies all users access to JOB/SCAN.
GROUPID=xxxxxxx	Grants specific groups of users access to JOB/SCAN. <b>(For RACF Only)</b>
USERID=xxxxxxx	Grants specific users or multiple users access to JOB/SCAN.
STANDARDS=NONE	Designates that none of the specified users can access the Standards Facility.
STANDARDS=xxxxxxx	Designates that the specified users must use the standards program named in this parameter.
STANDARDS=ANY	Designates that the specified users can specify a standards program at run time.
PARM='stdsparm'	Specifies an optional parameter that is passed to the standards program. The maximum is 32 characters enclosed in quotations.
PARM=ANY	Allows the specified users to enter an optional parameter for the specified standards program. If STANDARDS=xxxxxxx is specified, this will allow

Parameter	Function
	users to specify a PARM for the required standards program.
REFCMDON	Allows users to code Reformatter commands (which will override any Bx Control Table entries). This is the default.
REFCMDOFF	Prevents users from using Reformat commands. Bx Control Table entries or Reformatter defaults will be used.
SUBMIT/ NOSUBMIT	Allows user to submit JCL to MVS after validation from the JOB/SCAN ISPF Dialog panel. Default is NOSUBMIT.
MEMBER=xxxxxxxx	Designates an assigned options member that overrides JSOPT03.

### JOB/SCAN Usage File Detail Statement

The JOB/SCAN Usage File detail statement enables writing product usage data to the usage file. See [Activating and the Product Usage Facility](#).

#### Format



#### Parameters

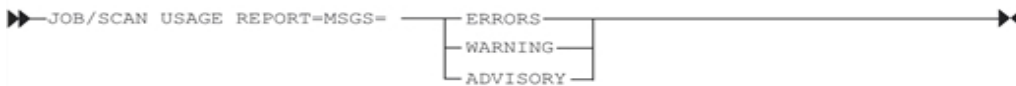
**YES** Default. Writes usage statistics to the usage file.

**NO** Prevents writing statistics to the usage file.

### JOB/SCAN Usage Report Detail Statement

The JOB/SCAN Usage Report detail statement enables product usage data gathering and allows the user to see product usage data. This detail statement must be present in the JSOPT01 member in the SJJSPARM library to enable data gathering.

#### Format



#### Example

```
JOB/SCAN USAGE REPORT=MSG=WARNING
```



## Parameters

### MSGs

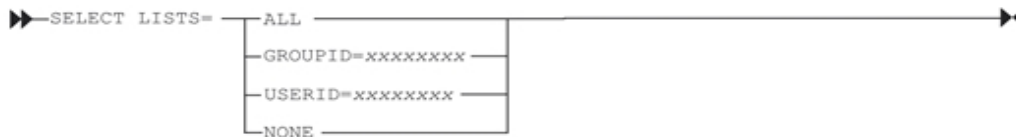
The MSGS parameters controls the types of errors included in the JOB/SCAN Usage Report. As the default, JOB/SCAN reports all Advisory, Warning, and Error messages.

Parameter	Function
ERRORS	Specifies only Error level messages be included in the Usage Report. This is the default setting.
WARNING	Specifies only Warning and Error level messages be included in the Usage Report.
ADVISORY	Specifies all levels of messages (Advisory, Warning, and Error) be included in the Usage Report.

### Select Lists Detail Statement

The Select Lists detail statement controls which users can access the Selection List Maintenance Facility panel from the ASG-JOB/SCAN Primary Menu panel. This facility allows users to create lists of jobs to be documented as a group.

### Format



### Example

```
SELECT LISTS=USERID=APL*****
```

### Parameters

See [Parameters](#) for additional information about entering userids and groupids.

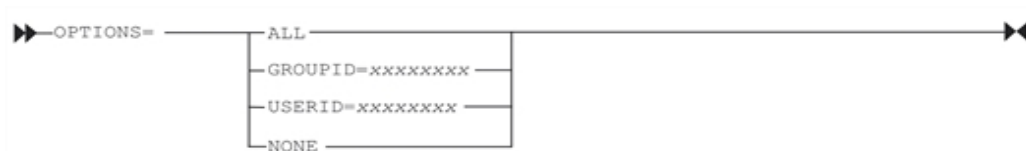
Parameter	Function
ALL	Grants all users access to the panel.
NONE	Denies all users access to the panel.
GROUPID=xxxxxxx	Grants specific groups of users access to the panel.
USERID=xxxxxxx	Grants specific users or multiple users access to the panel.

### Options Detail Statement

The Options detail statement controls which users can access the Assign Alternate Options Members panel from the ASG-JOB/SCAN Primary Menu panel. Users access this panel to designate personal options members and to display default

options member names and any alternate options members assigned.

### Format



### Example

```
OPTIONS=NONE
```

### Parameters

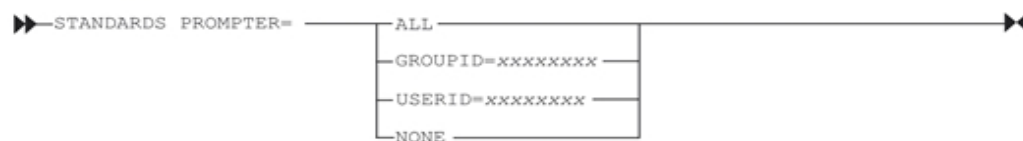
See [Parameters](#) for additional information about entering userids and groupids.

Parameter	Function
ALL	Grants all users access to the panel.
NONE	Denies all users access to the panel.
GROUPID=xxxxxxx	Grants specific groups of users access to the panel.
USERID=xxxxxxx	Grants specific users or multiple users access to the panel.

### Standards Prompter Detail Statement

The Standards Prompter detail statement controls which users can access the Standards Utility Menu panel from the ASG-JOB/SCAN Primary Menu panel. This facility allows users to create Standards programs using the Standards Prompter. To add utilities to the Standards Utility Menu panel, see [Adding Items to the Standards Utility Menu Panel](#).

### Format



### Example

```
STANDARDS PROMPTER=USERID=APL*****
```

### Parameters

See [Parameters](#) for additional information about entering userids and groupids.

Parameter	Function
ALL	Grants all users access to the panel.
NONE	Denies all users access to the panel.
GROUPID=xxxxxxxx	Grants specific groups of users access to the panel.
USERID=xxxxxxxx	Grants specific users access to the panel.

### User Function Detail Statement

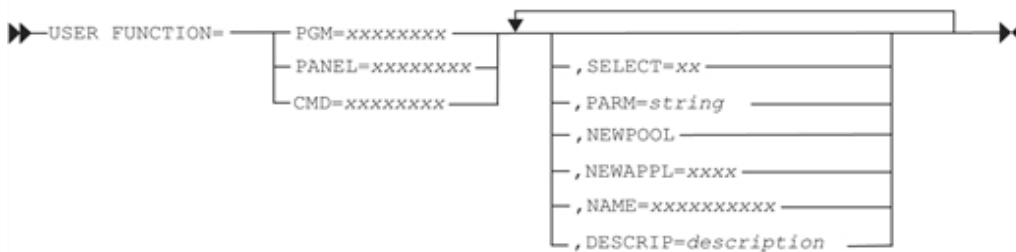
The User Function detail statement controls access to functions that are not part of JOB/SCAN, but which may be made available from the ASG-JOB/SCAN Primary Menu panel. The Product Administrator can add up to three options to the ASG-JOB/SCAN Primary Menu panel, giving users access to other site applications, such as EDIT or BROWSE functions or utilities. See [Adding Site-Defined Functions](#).

Because there are no ALL, NONE, or USERID= parameters for the User Function detail statement, the function is available to every user.



If the region requirements of any user functions exceed 1 MB, it might be necessary to increase the overall region size beyond the normal 2 MB for JOB/SCAN.

### Format



### Parameters

Parameter	Function
PGM=xxxxxxxx	Specifies program to be invoked whenever this user function is selected from the ASG-JOB/SCAN Primary Menu panel.
PANEL=xxxxxxxx	Specifies ID of the to-be-displayed panel whenever this user function is selected from the ASG-JOB/SCAN Primary Menu panel.
CMD=xxxxxxxx	Specifies command to be executed whenever this user function is selected

Parameter	Function
	from the ASG-JOB/SCAN Primary Menu panel.
SELECT= <i>xx</i>	Specifies the code JOB/SCAN displays on the ASG-JOB/SCAN Primary Menu panel to invoke this user function. The code can be one or two characters and is displayed on the panel. It must not be 0 to 15, X, or any valid ISPF command such as UP. Defaults for this field are U1 and U2 (one for each of the two possible user functions).
NEWPOOL	(Optional) Specifies that a new ISPF shared-variable pool is to be created for this function without specifying a new application ID.
NEWAPPL= <i>xxxxx</i>	(Optional) Specifies the ISPF application ID for this user function.
NAME= <i>xxxxxxxxxxxx</i>	(Optional) Specifies the name of the function, up to 10 characters. JOB/SCAN displays the name on the ASG-JOB/SCAN Primary Menu panel to the left of the dash. If the name includes blanks or commas, enclose it in single quotes.
DESCRIP= <i>'description'</i>	(Optional) Specifies the description of the function. JOB/SCAN displays the description on the ASG-JOB/SCAN Primary Menu panel (to the right of the dash). Maximum length of the field is 38 characters; enclose it in single quotes. Default for this field is 'User Function n' where n is 1, 2 or 3 depending on the position of the statement.
PARM= <i>'string'</i>	(Optional) Specifies up to 24 characters of parameter information to be passed to the user function; enclose the data in single quotes.

## File Allocations Group

The detail statements comprising the File Allocations group concatenate dataset names to JOB/SCAN ddnames. Default File Allocations statements are delivered in the JSOPT02 member in the SJJSPARM library. All File Allocations statements can be modified or changed using the parameters described in this section. Files specified in these statements are only allocated when needed to conserve file allocation space for JOB/SCAN.

### Allocating Files Using the TSO ALLOC Command

In ISPF mode, use the TSO ALLOC command to allocate the files. In batch mode, specify DD statements to allocate the files. If a file has already been allocated and is also defined in a File Allocations statement, JOB/SCAN ignores the File Allocations statement.

#### Format

File Allocation statements use dd names as keywords; as such, they do not use the usual detail statement syntax. Put concatenated libraries on separate lines, with a leading equal-sign, and no ending comma. To concatenate datasets,

specify multiple dataset names with a single ddname.

### Example

For example, to concatenate multiple libraries for the JLJPA File Allocation statement:

---

```
FILE ALLOCATIONS
  JLJPA  =SHR, DSN=SYS1.PROCLIB
        =SHR, DSN=SYS2.PROCLIB
        =SHR, DSN=SYS3.PROCLIB, UNIT=SYSDA, VOLSER=TEST01
        =SHR, DSN=SYS3.PROCLIB, UNIT=SYSDA, VOLSER=PROD01
```

---



UNIT and VOLSER parameters are required only for uncataloged datasets.

### DDnames used with File Allocation Statements

This table describes ddnames that can be allocated with File Allocation statements in JSOPT02. These names will be dynamically allocated *if the DDNAME has not been pre-allocated via JCL or a TSO ALLOCATE command*.

DDname	Function
ACBLIB	Required for IMS. Enables JOB/SCAN to find necessary IMS database information for BMP jobs.
DAGLOBAL	Required by CONTROL-M setup and Auto-Edit interface. Contains Auto-Edit variables.
JLDIAG	Output dataset for diagnostics; usually specified as Dummy, or as requested by ASG.
JLINTRDR	Disk dataset used for submit processing.
JLJPA	PROCLIB datasets which JOB/SCAN accesses when validating JCL. This statement is modified during installation. (See <a href="#">Step 2—Specify PROCLIBs in JSOPT02</a> )
JLJPF	Designates an optional control card library to be used when the one specified in JCL needs to be overridden, accessed by JOB/SCAN at run time. However, it is still necessary to include an entry in Control Table AB referencing ddname to be overridden, or an entry in Table AQ referencing dataset name to be overridden. Access Control Cards Run Option must be set to YES.
JLJPWOE	File used for <i>Test an Options Member</i> function on Product Administration panel.
JLJPWJL	PDSOUT used for Structured JCL Listing.
JLLAA	Required when using submit-after-checking in ISPF. Also required if Write Selected JCL Run Option is set to YES in batch mode.
JLSCHR	Contains the resolved JCL from the scheduler interface. The use of this DD is fully explained in <a href="#">User Information - JOB/SCAN with Schedulers</a> .

DDname	Function
JLSCRN	Used for Structured JCL display for ISPF.
JLSCRN2	Used for JCL Change and Reformatter output for ISPF.
JLUTL01	Specifies a utility DSN used by Reformatter and Change for temporary disk storage.
PRTDBG	Required by CONTROL-M setup and Auto-Edit interface. Used by internal CONTROL-M debugging routines. File may be defined as DUMMY.
SYSOUT	Contains output of JSVERIFY, displaying all JCL parameter values passed to the Standards Facility.

### DDnames Allocated Without Using File Allocation Statements

Some ddnames are allocated without using File Allocation statements. Instead, they are coded in either the logon PROC, reallocate CLIST, or batch PROC, depending on how JOB/SCAN was installed.

Other ddnames are dynamically allocated by JOB/SCAN as the product executes. For example, the PROC library name entered in the First PROCLIB field of the JOB/SCAN JCL Validation panel is associated with ddname JLJPC; that ddname does not need a File Allocation statement.

This table show the DDnames allocated without using File Allocation statements.

DDname	Function
JLCHLI	Assigned as JCL Change input - printed output (listing), only used in batch.
JLCHLO	Assigned as JCL Change output (listing).
JLCHMO	Assigned to the library where members that have been changed or reformatted will be written. Dynamically allocated if the Write Changed JCL? field is set to Y and the Output Lib field is filled in on the Change/Reformat panel. May also be coded in batch execution JCL as a DD statement.
JLJOB	Used for Input Selection. Dynamically allocated as a result of a DSN= Input Selection statement. Overridden by a DDNAME= statement. May also be coded in batch execution JCL as a DD statement.
JLJPB	Allocated if EZPROCLIB parameter is turned on in Program Product Interface Run Option and a //PROCLIB DD, //JOBPROC DD, //DYNAPROC DD, or //JCLLIB ORDER statement is encountered. Specifies libraries to be searched for PROCs before searching JLJPA libraries.
JLJPC	Dynamically allocated if First PROCLIB field is filled in on the JCL Validation panel. This ddname specifies PROCLIBs to be searched first before JLJPA and JLJPB datasets. JLJPC can also be specified in a batch PROC to indicate first file to be searched for PROCLIBs. If a PROCLIB is not

DDname	Function
	found in JLJPC, JLJPB is then searched. (See <a href="#">Run Options Group</a> for an explanation of the JLJPC PROCLIB Type Run Option enabling the specification of a PROCLIB type other than PDS (the default)).
JLKAC	Assigned as hardcopy printout of the Structured JCL Listing.
JLLOD	Not required unless an ISPEXEC LIBDEF ISPLLIB statement is being used to allocate load library. In this case, JLLOD should be allocated in a logon PROC or CLIST.
JLMSG	DDname that points to JOB/SCAN extended error message text (batch only).
JLOPE	Assigned as hardcopy printout of Options In Effect report and Table of Contents report. Dynamically allocated by JOB/SCAN.
JLRCL	Assigned to the SJJSPARM library. Parameters for this ddname are usually coded in the logon PROC, allocation CLIST, or batch PROC, depending on how the product is installed.
JLRULES	Assigned to the library containing the JCL Change or Reformatter commands. Dynamically allocated if the Command Lib field is filled in on the Change/ Reformat panel. May also be coded in batch execution JCL as a DD statement.
JLSCH	DDname containing the scheduler control cards. The use of this DD is fully explained in the <i>JOB/SCAN Scheduler Interface Guides</i> .
JLSCHO	Listing containing the scheduler control cards and any associated errors. The use of this DD is fully explained in the <i>JOB/SCAN Scheduler Interface Guides</i> .
JLSNAP	Contains JOB/SCAN diagnostic information.
JLSSR	Holds system-level messages generated during the execution of JOB/SCAN. In foreground, JLSSR is usually allocated to the terminal and it is dynamically allocated when needed. There is no need to preallocate JLSSR for foreground JOB/SCAN work. When running JOB/SCAN in batch, JLSSR must be allocated via a DD statement pointing to a sysout class (SYSOUT=*) or a data set.
JLSUMR	Assigned as JCL Change Summary Report.


### Parameters

This table describes the valid parameters for File Allocation statements and associates some parameters with parameters for DD statements from JCL.

File Allocation Parameters	Related DD Parameters	Function
&UID	None	Subparameter for DSN= parameter. Inserts userid into DSN parameter.
&GRP	None	Subparameter for DSN= parameter. Inserts groupid into DSN parameter.
&PF	None	Subparameter for DSN= parameter. Inserts TSO PREFIX into DSN parameter.
AVGREC=	AVGREC	When BLKS parameter is coded, AVGREC specifies that primary and secondary space quantities coded in PRIM and SEC parameters represent units (u), thousands (k), or millions (m) of records. When BLKS parameter is not coded, this parameter is ignored.
BLKS	SPACE	Specifies unit of allocation for a new dataset is blocks.
BLKSIZE=	BLKSIZE	Specifies block size to be used when allocating a file. If BLKSIZE=0 is coded, to have the system determine the block size, the LRECL, RECFM, and DSORG parameters must be coded.
CATLG	CATLG	Specifies new dataset is to be cataloged.
CHARS=	CHARS	Specifies character set to be used for SYSOUTs.
COPIES=	COPIES	Defines number of copies to be generated for SYSOUTs.
CYLS	SPACE	Specifies unit of allocation for a new dataset is cylinders.
DATACLAS	DATACLAS	Specifies name of a data class to be used for allocating dataset. Name (one to eight characters) is defined by the storage administrator at the installation site.
DELEND	DISP	Specifies file to be deleted at end of processing, if it was created at the beginning of processing.
DELETE	DELETE	Specifies file to be deleted when no longer needed.
DEST=	DEST	Specifies JES destination for SYSOUTs.
DIR=	SPACE	Specifies the number of directory blocks used when allocating a new PDS.
DSN=	DSN	Identifies dataset name associated with ddname.



File Allocation Parameters	Related DD Parameters	Function
DSORG=	DCB	Specifies dataset organization when allocating a file.
DUMMY	DUMMY	Sets DD to DUMMY (DSN=NULLFILE).
FCB=	FCB	Specifies name of forms control buffer image that JES will use when printing of this SYSOUT dataset.
FORMDEF=	OUTPUT	Identifies a library member that contains statements to tell Print Services Facility (PSF) how to print SYSOUT dataset on a page-mode printer. This parameter is not supported on MVS releases prior to SP 3.1.
FORMS=	FORMS	Specifies the SYSOUT forms to be used for this file.
FSSDATA=	OUTPUT	Specifies a subsystem-defined parameter to pass from a spooling product to a functional subsystem application (FSA) or other de-spooler. If the parameter contains special characters, enclose the parameter in apostrophes.
HOLD	HOLD	Specifies that SYSOUT is to be placed in the hold queue.
INTRDR	SYSOUT	Routes JCL to internal reader.
KEEP	KEEP	Specifies that dataset is to be kept after JOB/SCAN run.
LRECL=	LRECL	Defines logical record length of a new file.
MBR=	DSN	Allocates a member of a library.
MGMTCLAS=	MGMTCLAS	Specifies name of a management class to be used for allocating the dataset. Name (one to eight characters) is defined by the storage administrator at the installation site.
MOD	MOD	Used to specify the creation of a new file, provided that the file does not already exist. If the new file exists, adds to the end of it.
NEW	NEW	Identifies file as new.
OLD	OLD	Identifies file as old.
PAGEDEF	OUTPUT	Identifies a library member that contains statements to tell Print Service Facility (PSF) how to print SYSOUT dataset on a page-mode printer.

File Allocation Parameters	Related DD Parameters	Function
		 This parameter is not supported on MVS releases prior to SP 3.1.
PRIM=	SPACE	Specifies amount of primary space to be allocated for a new dataset. BLKS parameter must also be specified.
RECFM=	RECFM	Specifies record format of file to be allocated.
REUSE	None	Specifies that if the DDNAME was previously allocated (by ASG-DOCU/TEXT, JOB/SCAN, or ASG-INFO/X), the DDNAME should be unallocated then re-allocated with the parameters specified on this FILE ALLOCATION statement.
SEC=	SPACE	Specifies secondary space to be allocated for a new dataset.
SHR	SHR	Specifies that this file can be shared.
STORCLAS	STORCLAS	Specifies name of a storage class to be used for allocating dataset. Name (one to eight characters) is defined by the storage administrator at the installation site.
SUBSYS=	SUBSYS	The SUBSYS parameter is used to request a subsystem to process this dataset and, optionally, to specify parameters defined by the subsystem.
SYSOUT=	SYSOUT	Defines SYSOUT class.
TERM	TERM	Designates that this DD is allocated to the terminal.
TRKS	SPACE	Specifies unit of allocation for a new dataset is tracks.
UCS=	UCS	Designates UCS to be used for SYSOUTs.
UNIT=	UNIT	Specifies unit name when allocating a new dataset.
VOLSER=	VOL=SER	Specifies volume serial number used when allocating a new dataset.



On the Hardcopy Output Characteristics panel, values entered for Class, Dest, and so on override their equivalent File Allocation values.

## Special Considerations for JES2 /\*JOBPARM or JES3 /\*\*MAIN Statements

If using JES2 /\*JOBPARM or JES3 /\*\*MAIN statements in jobs to be validated, follow these rules.

- For each /\*JOBPARM PROCLIB= statement in JES2 and each /\*\*MAIN PROC= statement in JES3, associate them with a unique JLJPA File Allocation statement.
- If the PROCLIB name is four to eight characters, use the actual PROCLIB name. For example, for PROC libraries identified as PROCLIB=TEST01 and PROCLIB=PAYROLL, add the File Allocation statements TEST01 and PAYROLL, respectively.
- If the PROCLIB names are one-, two-, or three-characters in the form of PROCLIB=xxx, JES2 statements can use the ddname JLJPAxxx, where xxx is the actual PROCLIB name.

For example, for the PROC libraries identified as PROCLIB=TEST01 and PROCLIB=XY,

---

```

/*JOBPARM Statement #1:
//PJMN0040 JOB (A,P,MN), 'SMITH', CLASS=P, MSGCLASS=H
/*JOBPARM PROCLIB=XY
//PP20 EXEC PAYPROC, INDX=D176, TAPEVOL=D98073, TYPE=D
//...
/*JOBPARM Statement #2:
//PJRST JOB (A,T,GL), 'DEPT. 72', CLAS=T, NOTIFY=DT100
/*JOBPARM PROCLIB=TEST01
//PSTEP1 EXEC GENLEDG, UNT=SYSDA, MBR=CNTRL1
//...

```

---

place the File Allocation statements TEST01 and JLJPAXY in JSOPT02:

---

```

FILE ALLOCATIONS
  JLJPA   =DSN=SYS1.PROCLIB,SHR           System default proclibs
          =DSN=SYS2.PROCLIB,SHR
          =DSN=SYS1.MISCPROC,SHR
          =DSN=SYS1.PROD.PROCLIB,SHR
  JLJPAXY =DSN=OPS.TEST.PROCLIB,SHR      File allocs for JOBPARM #1
          =DSN=PC.TEST.PROCLIB,SHR
  TEST01  =DSN=USER.TEST.PROCLIB,SHR     File allocs for JOBPARM #2
          =DSN=APPS.DEVL.PROCLIB,SHR

```

---

## Run Options Group

The detail statements comprising the Run Options group control how JOB/SCAN processes the JCL. They provide the ability to customize various aspects of JCL validation. Run Options statements can reside in any member of the SJJSPARM library (usually not in JSOPT01).

For complete information about all JOB/SCAN's Run Options, see [Run Option Statements](#).

### Format and Example

When specifying Run Options statements, begin with the Run Options group-level statement and follow it with as many detail-level statements as necessary. The statement group can appear anywhere in the SJJSPARM library member, provided the group-level statement immediately precedes the detail-level statements.

#### Sample Run Options statement

```

RUN OPTIONS
  GROUP ID=GROUPID
  USER ID=USERID
  JOB LIMIT=99998
  JSP=JS6PROC

```

## Detail Statements

Run Option statements control the processing of information during a JOB/SCAN run and affect the contents of all reports. For example, you can:

- Change the page length of the report.
- Limit the number of control cards printed in the Structured JCL Listing.
- Request special IMS processing.
- Turn on IDCAMS processing.
- Request condition code simulation.

## Overriding Selected Run Options at Run Time

The values entered in the JOB/SCAN panels override some Run Options and Report Options coded in the SJJSPARM library.

To run JOB/SCAN in batch, code Run Options statements directly in the options members using any editing method in the SYSIN stream in the JCL.

## Adding or Modifying Run Option Statements

### *To add new Run Option statements or modify the parameters of existing statements*

1. Select the Product Administration Menu panel from the ASG-JOB/SCAN Primary Menu panel. The Product Administration panel appears.
2. Select Modify Options Members from the Product Administration panel. A directory of the entire SJJSPARM library appears.
3. Select the appropriate member from the directory. The member is displayed in ISPF Edit mode.
4. Add or change Run Options as required.
5. Test the additions or modifications. See [Testing an Options Member](#).

## Determining Run Options in Effect

This table describes the commands that can be used on any JOB/SCAN panel to quickly determine out which Run Options are in effect. (JOB/SCAN help panels are not valid environments for these commands.) [Portion of a SHOW RUN command Run Options Summary](#) shows a portion of the Run Options summary.

Command	Description
<b>SHOW OPTIONS</b>	View the Run Options, Report Options and Control Tables in effect.
<b>SHOW RUN</b>	View the Run Options only.

### Portion of a SHOW RUN command Run Options Summary

```

----- SUMMARY OF OPTIONS IN EFFECT ----- Row 1 to 38 of 65
COMMAND ==>                                SCROLL ==> PAGE
*****
RUN OPTIONS
*****
ACCESS CONTROL CARDS      =YES, LIMIT=000
CATALOG ACCESS           =YES, CVOL, INTCATB, PDSMEM, PGM, VTOC
CONTROL-M VARIABLES     =NORESOLVE
CONTROL-M EXITS         =CTMX002, CTMSE02
DBCS                     =NO
. . .
JOB LIMIT                =
JOB/SCAN PROC            =JS6PROC
. . .

```

## Report Option Group

The detail statements comprising the Report Option group control the contents and layout of JOB/SCAN reports. Report Option statements can reside in any member of the SJJSPARM library (usually not in JSOPT01).

### Example

Sample Report Options detail statements in an SJJSPARM library member

```

REPORT OPTIONS
  OPTIONS IN EFFECT =PRINT, SUMMARY, WIDTH=132
  JCL CHANGE REPORT =INJCL, COMMANDS, OUTJCL, SUMMARY
  STRUCTURED JCL LISTING=ERRORSONLY, WIDTH=132

```



When executing JOB/SCAN from the panels, the report type and type of media (Hardcopy or Display (Browse or View)) are specified on the JOB/SCAN - JCL Validation panel. The values used on the panel override the same parameters coded in the SJJSPARM library members.

### Report Options Detail Statements

This table summarizes the Report Option detail statements used in JOB/SCAN.

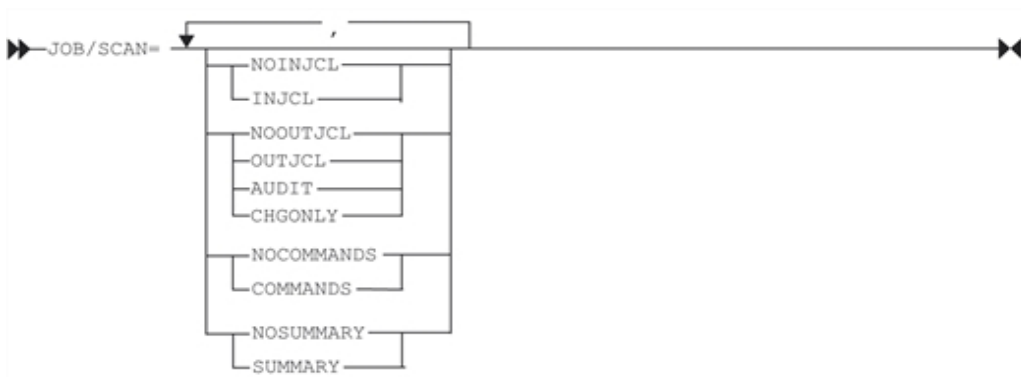
Keyword	Purpose
JCL Change Report (JCR)	Provides options for producing input and output reports for JCL Change and/or Reformatter.
Options in Effect (OPE)	Produces a report showing the Control Tables and control statements used in JOB/SCAN.
Structured JCL Display (SJD)	Displays the Structured JCL Listing at a terminal.

Keyword	Purpose
Structured JCL Listing (SJL)	Produces a report showing the fully resolved JCL. This Report Option controls whether JCL error messages will be included in the detail of the report, whether the Space Analysis will be printed, and the width of the report.

**JCL Change Report (JCR)**

This detail statement controls input and output listings in the JCL Change and Reformatter reports. Without this statement, JOB/SCAN generates no report.

**Format**



**Example**

JCR=INJCL, AUDIT

**Parameters**

Parameter	Default	Purpose
NOINJCL/INJCL	NOINJCL	Lists input JCL.
NOOUTJCL/ OUTJCL/ AUDIT/ CHGONLY	NOOUTJCL	Lists output JCL. JOB/SCAN places \$ in column 1 to mark statements changed, showing JCL exactly as finished, or Audit report showing the JCL before and after the run. An O (old), N (new), D (delete), A (add), or blank for no change is placed in column 1 of the Audit Report. The CHGONLY parameter specifies that only the changed statements, along with their corresponding original JCL statements, are included. A C (changed) or O (original) is placed in column 1.
NOCOMMANDS/	NOCOMMANDS	Listing of the Change and/or Reformatter commands used to

Parameter	Default	Purpose
COMMANDS		process the JCL.
NOSUMMARY/ SUMMARY	NOSUMMARY	Report giving summary of changes by type of change. Column headings on the report are: <ul style="list-style-type: none"> <li>• STATUS 1, 2, 3 (see Status table below)</li> <li>• #STMTS IN</li> <li>• #STMTS OUT</li> <li>• #STMTS CHGD</li> <li>• #STMTS ADDED</li> <li>• #STMTS DELETED</li> </ul>

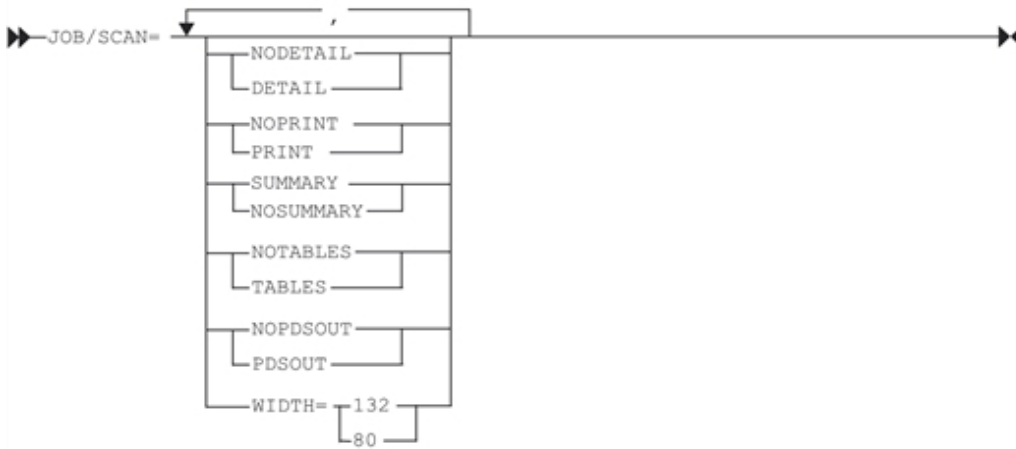
This table lists the meanings of the output in the STATUS 1 2 3 column.

STATUS			Description
1	2	3	
-			Change was not requested.
*			No changes were made to member.
C			Changes were made to member.
E			An error was found.
	-		Reformat was not requested.
	*		No statements were reformatted.
	R		Member was reformatted.
		-	Output was not requested.
		*	Output member was not written (WCO=YES).
		W	Output member was written.

### Options in Effect (OPE)

This detail statement produces a report showing the Control Tables and control statements used in JOB/SCAN execution.

**Format**



**Example**

`OPE =PRINT,DETAIL, TABLES`

**Parameters**

Parameter	Default	Purpose
NODETAIL/DETAIL	NODETAIL	Specifies whether or not to produce a report containing all control statements, except Control Tables, as read from the SJJSPARM library members.
NOPDSOUT/PDSOUT	NOPDSOUT	Specifies whether or not to write the report to a PDS output file identified by the JLPWOE DD. The run-time options and the table information - if requested - is written to the OPTIONS member.
NOPRINT/PRINT	NOPRINT	Specifies whether or not to print the report.
SUMMARY/ NOSUMMARY	SUMMARY	Specifies whether or not to produce a summary of the options actually used during JOB/SCAN execution.
NOTABLES/TABLES	NOTABLES	Specifies whether or not to list the Control Tables.
WIDTH	132	Specifies the width of the report; allowable widths are 132 and 80.

**Structured JCL Display (SJD)**

This detail statement displays the Structured JCL Listing on the user's terminal screen. The File Allocation ddname JLSCRN designates the display's output. This run option does not apply to the JEM Edit macro.



## Format



## Example

SJD=ERRORSONLY

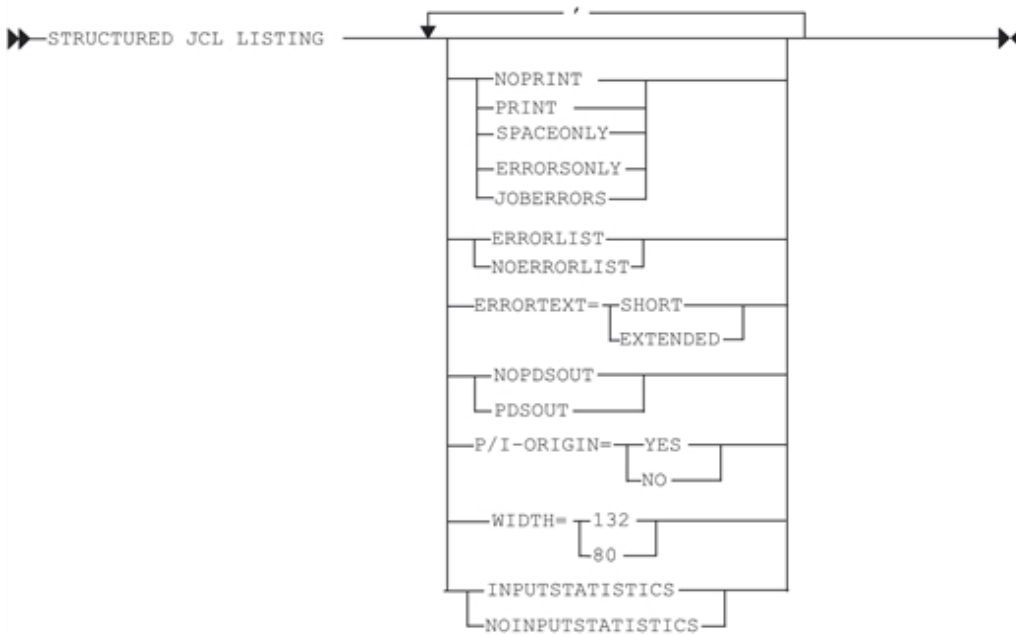
## Parameters

Parameter	Default	Purpose
YES/	NO	YES displays the Structured JCL Listing (with the Error Summary).
NO		NO displays the Error Summary only if there are errors.
ERRORSONLY		ERRORSONLY displays the Error Summary on the screen, regardless of error state.

## Structured JCL Listing (SJL)

This detail statement produces a report showing a fully-resolved JCL listing. This statement controls whether JCL error messages will be printed in the detail of the report, whether the Space Analysis will be printed, and the width of the report. The value entered on the JOB/SCAN - JCL Validation panel overrides the NOPRINT/PRINT/SPACEONLY/ERRORSONLY/JOBERRORS parameter coded in this statement.

## Format




## Example

```
SJL =PRINT,P/I-ORIGIN=Y
```

## Parameters

Unless otherwise noted, these parameters do *not* apply to the JEM Edit macro.

Parameter	Default	Purpose
NOPRINT/ PRINT/ SPACEONLY/ ERRORSONLY/ JOBERRORS	NOPRINT	Specifies whether or not to print the hardcopy report: SPACEONLY prints only the Space Analysis. ERRORSONLY prints only the Error Summary. JOBERRORS prints the Error Summary with a page break for each job scanned.
ERRORLIST/ NOERRORLIST	ERRORLIST	Specifies whether or not JOB/SCAN error messages are printed following the statements in error. NOERRORLIST suppresses printing of the error messages within the detail of the report. The Error Summary is still generated; it summarizes error messages and references the appropriate statement, index number, and page where the error occurred.

Parameter	Default	Purpose
ERRORTXT	SHORT	<p>Controls generation of error messages on the Structured JCL Listing:</p> <p>SHORT—specifies that only the standard single-line error message will be printed following a statement in error.</p> <p>EXTENDED—specifies that extended error text from DSSMSG00 will be automatically included in the Structured JCL Listing following the short error message.</p> <p>When running JOB/SCAN from the ISPF panels, this option can be entered on the Processing Options panel rather than by coding the Run Option parameter. The value entered on the panel overrides any corresponding parameter coded in the SJJSPARM library.</p>
P/I-ORIGIN=YES/NO or PIO=Y/N	NO	<p>Specifies whether JOB/SCAN issues informational messages DSS2000 (where procedure was expanded from), DSS2001 (where include group was expanded from) and DSS2002 (the library in which the procedure to include group was first found). This parameter applies to the JEM macro.</p> <p>P/I-ORIGIN may be abbreviated to PIO</p> <p>YES may be abbreviated to Y</p> <p>NO may be abbreviated to N</p>
NOPDSOUT/ PDSOUT	NOPDSOUT	<p>Specifies whether JOB/SCAN writes the report to a PDS output file identified by the JLPWJL DD. The detail section of the report is stored under the job name. The summary section is stored in the member called SUMMARY. The input statistics report is written to the RUNSTATS member.</p> <div style="border: 1px solid purple; padding: 10px; margin-top: 10px;"> <p> PDSOUT is an ASG-DOCU/TEXT function and requires the Extended Functions feature of ASG-DOCU/TEXT to be in the JOB/SCAN product authorization code. Contact your Product Administrator or ASG customer support for assistance.</p> </div>
WIDTH	132	Determines width of the report. Allowable widths are 132 and 80.

Parameter	Default	Purpose
INPUTSTATISTICS/ NOINPUTSTATISTICS	INPUTSTATISTICS	Specifies whether JOB/SCAN prints the Input Statistics report at the end of the SJL.

## Control Table Group

The detail statements comprising the Control Table group (herein called Control Tables) provide site-operational details that guide JOB/SCAN's processes JCL. For example, use a Control Table to identify specific ddnames that reference control card libraries. Control tables also control the Reformatter. See [Control Tables](#) for detailed descriptions.

Each control table is uniquely identified by a two-character code entered in columns 6 and 7. Control Table values are set at installation time and can be modified as necessary.

### Accessing Control Tables

Control Tables reside in the JSOPT02 or JSOPT03 members in the SJJSPARM library. This member can be modified as necessary to reflect exact details for the user's site. Bring member into Edit to modify tables.

## Default Control Tables in JSOPT02

```

----- Advanced PRO/JCL Installation Settings -----
Command ==>
More:      +
----- Started Task Instance Information -----
Started Task ID (blank is recommended):
==> 0

SYSOUT Class for the Started Task:
==> *

Enable Directed Execution requests to this started task? (Y/N):
==> Y

Enable WEB Based requests? (Y/N):
==> Y

Enable Batch Usage Statistics? (Y/N):
==> Y

----- Configure Network Affinity (optional) -----
If your site requires network affinity enter the TCPIP stack name:
==>

----- Data Set Information -----
Are data sets to be SMS Managed? (Y/N):
==> Y

Storage Class (If SMS managed):
==> QAL

Unit Name (If non-SMS managed):
==>

Volume (If non-SMS managed):
==>

----- Language Environment Information -----
Are the Run-Time libraries in the LINKLIST (Y/N):
==> Y

Are the Run-Time libraries APF authorized (Y/N):
==> Y

High-Level-Qualifier for the Run-Time libraries:
==> CEE

Note: The High-Level-Qualifier for the Run-Time libraries must
be specified if they are not in LINKLST or APF authorized.

----- CA-7 Information -----
Generic CA-7   User ID: ==>           Password: ==>

Parm Passed to Batch CA-7 program SASSBSTR:
==>

Working CA-7 JCL Library      Index number: ==>
DSName: ==>

CA-7 Command File           DD Name:      ==> CA7CMDS
DSName: ==>

Batch Terminal Input        DD Name:      ==> BTERM0@I
DSName: ==> NULLFILE

Batch Terminal Output       DD Name:      ==> BTERM0@O
DSName: ==> NULLFILE

```

```

----- IBM Workload Scheduler Defaults -----
Default Subsystem name: ===>          (required if IPAddress is specified)

Default Remote System IPAddress if desired:
===>

Default Remote System Port if desired:
===>

***** END OF SETUP *****

```

Although Control Table entries are usually stored in JSOPT02, they can also be located in other SJJSPARM library members or in SYSIN data. The first value of a control table encountered always takes precedence.

### Control Tables with Associated Run Options

Some Control Tables have an associated Run Option statement that must be coded in order to use the table. The Run Option statement allows users to control the way in which a Control Table affects JOB/SCAN operation. The descriptions of Control Tables in the remainder of this section include references to such Run Options.

Control Tables may be modified by accessing JSOPT02, and changing the appropriate entries.

### Determining Control Tables in Effect

This command, used on any JOB/SCAN panel, displays the current Control Tables in effect.

**SHOW TABLES** An ISPF command that can be used on any JOB/SCAN panel to view a listing of the control tables in effect. (JOB/SCAN help panels are not valid environments for this command.)



**Determining Run Options in Effect** describes the command to determine which Run Options are in effect.

Figure shows a portion of the Control Tables summary.

### Result of a SHOW TABLES command

```

----- SUMMARY OF OPTIONS IN EFFECT ----- Row 1 to 40 of 92
COMMAND ===>                                SCROLL ===> PAGE

*****
CONTROL TABLES
*****

-----
|  AA  CONTAINS THE DEVICE NAMES IN USE AT THIS SITE FOR UNIT= VALUES. THE |
|      ONE-DIGIT CODE IS USED TO GENERATE FLOWCHART SYMBOLS.  CODES ARE:  |
|      R-PRINTED REPORT                                           |
|      D-DISK                                                       |
|      T-TAPE                                                       |
|      C-CARDS                                                       |
|      G-GRAPHICS OR CRT DEVICES                                   |
|-----|
AA CART36                                K
AA DISK                                  D          DEFAULT NAME FOR DAS
...

```

## Input Selection Group

The detail statements comprising the Input Selection group specify the libraries and members to be validated by JOB/SCAN in batch execution. Though Input Selection statements can be written in an SJJSPARM library member (JSOPT02), they are usually written in the SYSIN DD stream in the batch execution JCL (that is, copies of JSMVS). See [Validating JCL in Batch](#) for complete details about running JOB/SCAN in batch.


Edit the SYSIN DD stream or SJJSPARM library member to add new Input Selection detail statements or modify existing statements. See [Detail-Level Statements Format Rules](#) when entering detail statements for Input Selection statements.

### Input Selection Statement Format

When writing Input Selection detail statements, write the Input Selection group-level statement first, then write the detail statements. For example:

```
INPUT SELECTION
      DSN=MY.DATA.LIBRARY
      SOURCE TYPE =PDS
```

This table summarizes the Input Selection detail statements used in JOB/SCAN.

Keyword	Purpose
SELECTION EXIT	(Optional) Specifies the user-written, selection exit program name.  <div style="border: 1px solid purple; padding: 5px; display: inline-block;">  If specified, this must be the first statement after the INPUT SELECTION group statement. </div>
DATA SET NAME	Identifies the library containing the JCL members to be validated.
SOURCE TYPE	Specifies the type of library that stores the JCL to be validated. Type library can be PDS, PANVALET, LIBRARIAN, LIBRARIAN-3, PROCLIB or sequential files.
SELECT	Specifies the members [from the JCL library specified by the DSN statement] to be validated.
EXCLUDE	Excludes the specified members from validation.
DDNAME	(Optional) Identifies the ddname that specifies the library containing the JCL members to be validated. This is useful if the JCL resides in several different libraries.

### Data Set Name (DSN)

This detail statement specifies the library containing the JCL to be validated during JOB/SCAN execution.

### Format

```
▶▶ DSN= datasetname ▶▶
```

### Example

```
DSN =MY.DATA.LIBRARY
```

### Parameters

Parameter	Default	Purpose
<i>datasetname</i>	NONE	Specifies the library containing the JCL to be validated.

### Source Type (STP)

This detail statement specifies the type of library associated by the Input Selection DATA SET NAME (DSN) parameter.

### Format

```
▶▶ SOURCE TYPE=
┌── PDS ────────────────────────────────────────────────────────────────────────────────────▶
├── PANVALET | PAN ───────────────────────────────────────────────────────────────────────────▶
├── LIBRARIAN | LIB ───────────────────────────────────────────────────────────────────────────▶
├── LIBRARIAN-3 | LIB-3 ───────────────────────────────────────────────────────────────────────▶
├── PROCLIB ───────────────────────────────────────────────────────────────────────────────────▶
└── SEQUENTIAL ─────────────────────────────────────────────────────────────────────────────────▶
```

### Example

```
SOURCE TYPE =PDS
```

### Parameters

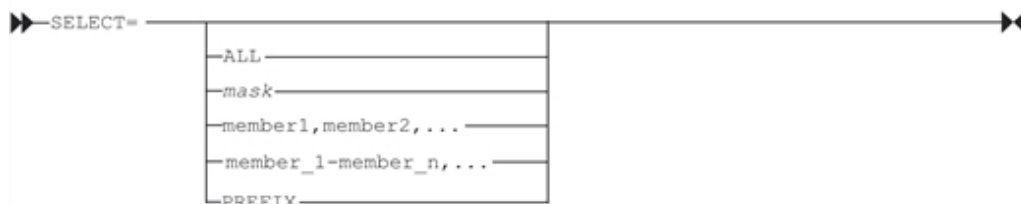
Parameter	Default	Purpose
LIBRARIAN/ LIBRARIAN-3/ PANVALET/ PDS/ PROCLIB/ SEQUENTIAL	PDS	Specifies the type of library associated by the Input Selection DATA SET NAME (DSN) parameter.

### Select (SEL)

This detail statement names the members in the JCL library to be validated during JOB/SCAN execution.

### Format





### Example

```
SELECT =ALL
```

### Parameters



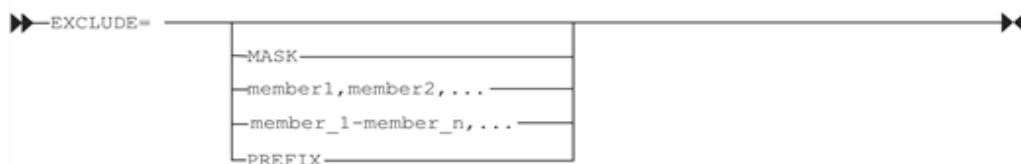
A member name of 'ALL' in the Select detail statement causes JOB/SCAN to load each member name in the PDS, regardless of any specified count.

Parameter	Description
<i>member_1,member_2,...</i>	One or more member names.
<i>member_1-member_n</i>	A range of similarly named members from low to high, in ascending order within the library.
<i>mask</i>	A mask for selection criteria. The asterisk is a wildcard character (matches anything) and can appear in any position.
PREFIX=	Prefix characters for member name selection. The prefix can be from one to seven characters.
ALL	All members in the current JCL library.

### Exclude (EXC)

This detail statement names the members in the JCL library to be excluded from validation during JOB/SCAN execution.

### Format



### Example

```
EXCLUDE =MYJOB
```

## Parameters

Parameter	Description
<i>member_1,member_2,...</i>	One or more member names.
<i>member_1-member_n</i>	A range of similarly named members from low to high, in ascending order within the library.
<i>mask</i>	A mask for selection criteria. The asterisk is a wildcard character (matches anything) and can appear in any position.
PREFIX=	Prefix characters for member name selection. The prefix can be from one to seven characters.

## DDNAME (DDN)

This detail statement identifies the ddname that specifies the library containing the JCL members to be validated during JOB/SCAN execution.

### Format

```
► DDNAME= ddname ◄
```

### Example

```
DDNAME =MYJLJOB
```

## Parameters

Parameter	Default	Purpose
<i>ddname</i>	JLJOB	Identifies the ddname that specifies the library containing the JCL members to be validated.

## Selection Exit (SELEX)

This detail statement identifies the optional, custom-written, selection exit program name. When specified, JOB/SCAN invokes the selection exit automatically for each JCL statement during JOB/SCAN execution.



If specified, this statement must be the first statement after the INPUT SELECTION statement.

### Format

```
▶▶SELECTION EXIT=——exitname——└┬,PARM=parm┘▶▶
```

### Example

```
SELECTION EXIT =CA7J2,'010920681730'
```

### Parameters

Parameter	Default	Purpose
<i>exitname</i>	NONE	Identifies the selection exit program name to be automatically invoked for each JCL statement during JOB/SCAN execution.
<i>parm</i>	NONE	Identifies the optional parameter to be passed to the Selection Exit program.

## JOB/SCAN Libraries

This topic describes the several libraries (datasets) that JOB/SCAN uses in setting up, controlling, and reporting the results of its scan processes. Other libraries contain members that interface to various third party software (that is, schedulers and data management software) or hold JCL useful for demonstrating aspects of JOB/SCAN.

- [SJSPARM Library](#)
- [SJSSAMP Library](#)
- [SJSSSTD Library](#)
- [SJSDemo Library](#)

### SJSPARM Library

The SJSPARM library contains options members that are used during JOB/SCAN execution. The options members contain statements and options, that:

- configure JOB/SCAN ISPF panels for users
- set up the JOB/SCAN run-time environment
- determine report information and formats
- specify file allocation information
- specify Reformatter settings

#### LIBRARY Allocation

The options members must reside in one or more libraries allocated to the ddname JLRCL. A logon PROC or CLIST and a batch PROC are set up during installation to allocate JLRCL.

Depending on how JOB/SCAN is configured at the user site, some members described in this documentation might not be used.

#### SJSPARM Members

The SJSPARM library contains options members JOB/SCAN reads during execution. Options members contain Access Control statements, Run Options, Report Options, and File Allocation statements used to tailor the JOB/SCAN environment. This table lists the default SJSPARM library members available after product installation.

Member Name	Description
DSSMSG00	Contains all JOB/SCAN extended error message text, including syntax, hidden, and run-time messages. Used for the Structured JCL Listing and JMSG Edit macro.
DSSMSG90	Includes all Standards Facility extended error message text. Used for the Structured JCL Listing and JMSG Edit macro.

Member Name	Description
JSOPTM	Contains the Input Selection, Run Option, and Report Option statements used by Edit macros. Can also specify the Standards Facility to be used with the Edit macros.
JSOPT01	Contains control information.
JSOPT02	Contains File Allocation statements.
JSOPT03	Contains Run and Report Option statements used when Validate JCL is selected from the ASG-JOB/SCAN Primary Menu panel and when Edit macros are used. Also contains user-supplied Control Table entries for the Standards Facility.
JSOPTRS	Required for using JOB/SCAN under ROSCOE/ETSO.

## Displaying the Directory of SJJSPARM Members

### To display a directory of SJJSPARM library members

- Select Modify Options Members from the Product Administration panel.

### Displaying SJJSPARM Members

```

EDIT      #USR.SJJSPARM                      Row 00021 of 00048
Command ==>
          Lib VV MM  Created      Changed      Size  Init  Mod  ID
. JSOPTM      1 60.25 93/09/02 96/01/10 10:23    21   14   0  ASGUS1
. JSOPTRS     2 60.02 93/09/02 95/11/20 09:47     9    9   0  ASGUS3
. JSOPTV2     2 60.02 93/09/02 96/01/22 09:49    82  117   0  ASGUS2
. JSOPT01     1 01.99 93/09/02 95/12/28 18:51    26   19   0  ASGUS5
. JSOPT02     1 01.49 93/09/02 96/03/13 13:39    98  114   0  ASGUS2
. JSOPT03     1 60.14 93/09/02 95/10/06 15:39    15   15   0  ASGUS3
. JS6PROC     1 01.48 93/06/23 96/02/11 16:33    34   28   0  ASGUS1
**End**

```



## SJJSSAMP Library

The SJJSSAMP library contains sample JCL for JOB/SCAN execution, third party interfaces, user exits, and other information. This table lists the members in the SJJSSAMP library.

Member	Description
ACF2EXIT	Security exit program for interfacing with CA-ACF2/MVS security package (provided for compatibility with security interfaces established for JOB/SCAN releases prior to 5.4). Does not support the dataset name validation function DSNAME (see the SAFx SJJSSAMP members for DSNAME support).

Member	Description
BETA42	JCL for interfacing with BETA42 Scheduler.
CASCHED	JCL for interfacing with CA Scheduler.
CASINTF	COBOL source code for interfacing with CA Scheduler.
CA7INTF	COBOL source code for interfacing with CA-7 Scheduler.
CA7J1	Sample selection exit that resolves CA-7 Scheduler #J cards given a single schedule-id.
CA7J2	Alternate sample selection exit that handles CA-7 Scheduler #J cards given multiple schedule-ids.
CA7SCHED	JCL that reads CA-7 Scheduler information, creates JOB/SCAN control statements for selected jobs, and executes DT5PROC.
CMN\$\$JCS	Example of ChangeMan ISPF skeleton, which invokes JOB/SCAN at JCL promotion.
CMN\$\$PRO	Example of ChangeMan ISPF skeleton, which invokes JOB/SCAN at procedure promotion.
CMN\$\$SCN	ISPF skeleton imbedded by CMN\$\$JCS and CMN\$\$PRO.
CONTROLM	JCL for interfacing with CONTROL-M Scheduler.
DCMSCHED	JCL for interfacing with DCMS Scheduler.
DCMSINTF	COBOL source code for interfacing with DCMS Scheduler.
ENDEVOR	ENDEVOR Processor for interfacing with ENDEVOR Scheduler.
ENQJS	Sample Assembler program for preventing a recursive run of JOB/SCAN.
ERRORMSG	Detailed JOB/SCAN error message text designed for online browsing.
ESPEVENT	This member is fully described in <a href="#">User Information - JOB/SCAN with ESP</a> .
ESPSCHED	This member is fully described in <a href="#">User Information - JOB/SCAN with ESP</a> .
INSJOB	Selection exit, which inserts a JOB statement into JCL members that do not begin with a JOB statement used primarily for validating stand-alone INCLUDE members.

Member	Description
JEMSUB	Edit macro to submit job streams for execution after an error-free validation.
JSMVS	JCL to run JOB/SCAN in an MVS batch environment. Executes JS6PROC.
JSMVSSUB	JCL to submit job streams for execution after an error-free validation.
JSVSAM	Member containing JCL used to create and define VSAM product usage file.
JSXINIT	Exit program used during initialization to obtain a userid or groupid to be used with the access control statements in the JSOPT01 member of the SJJSPARM library.
JSXTERM	Exit program used during termination.
JS6PROC	PROC invoked by JSMVS during JOB/SCAN execution in MVS batch environment.
JTRACSCH, JTRC2XIT, JTSCHED, JTSELECT	Sample COBOL programs and execution JCL which provide JOBTRAC scheduler interface.
J00YAAI	Sample JCL to rename load modules J00YAAI and J00YAAIA in order for JOB/SCAN to perform APF-authorized functions under ISPF.
J02YJS	Sample JCL to rename load module J02YJS in order for JOB/SCAN to perform APF-authorized functions in a Batch environment.
LINKPAM	JCL for linking JOB/SCAN with CA-PANVALET Release 11 or earlier.
PFORMAT	A sample job to format one or more collection periods from a product usage file.
PMAINT	A sample job to create or delete product usage file collection periods.
RACFEXIT	Security exit program that interfaces with RACF and CA-TOP SECRET security packages. Provided for compatibility with security interfaces established for JOB/SCAN releases prior to 5.4. Does not support dataset name validation function DSNAME. (See SAFx SJJSSAMP members for DSNAME support.)
READDSN	Assembler module (called by SJJSSAMP member JTRC2XIT) for interfacing with JOBTRAC scheduler.
SAFO	Security exit program that uses System Authorization Facility (SAF) to interface with RACF, CA-ACF2/MVS, and CA-TOP SECRET. This is the only level of SAF that can be used when running an ACF2 release prior to 6.

Member	Description
	<div style="border: 1px solid purple; padding: 5px;">  This program should not be used when running APF-authorized under ISPF with DSNNAME (coded in the Security Exit Run Option).         </div> <p>Security exit program that uses third-party RACHECK (instead of RACINIT/RACHECK, used by SAF0). Works with RACF release 1.8 or greater, CA-ACF2/MVS release 6 or greater, or equivalent CA-TOP SECRET release.</p>
SAF1	<div style="border: 1px solid purple; padding: 5px;">  SAF1 uses subpool 251 for storage when invoked in an APF-authorized environment under ISPF. Subpool 0 storage is deleted after program module is executed, while subpool 251 storage is not deleted until job set terminates.         </div>
SAF2	SAF2 performs same as SAF1 except it uses STATUS=ACCESS parameter of RACROUTE macro to eliminate logging of RACF violations (when user doesn't have proper access authority).
TWSSCHED	This member is fully described in <a href="#">User Information - JOB/SCAN with TWS</a> .
TWSSTD1	This member is fully described in <a href="#">User Information - JOB/SCAN with TWS</a> .
ZEKEDOC	Text member which provides examples of DD statements that must be added to ZEKE to allow JOB/SCAN to be executed from ZEKE. Both JCL and TSO allocate examples are provided.
ZEKESELC	Assembler module for interfacing with the ZEKE scheduler.
ZEKEVARS	Selection exit which interfaces with the ZEKE Scheduler to resolve ZEKE \$ global variables.
ZEKE14E	Assembler module, which contains a ZEKE User Exit 14. Allows ZEKE Scheduler to invoke JOB/SCAN for each job (before ZEKE submits it).

## SJSSTDL Library

The SJSSTDL library contains members that can be used with the JOB/SCAN Standards Facility:

- Sample source code for Standards programs. (Product Administrator: use this sample code to tailor programs for the local environment.)
- Copy members used by Standards programs.



The SJSSTDL library members, listed in this table, can be used with JOB/SCAN. (Depending on how this library is installed, it might not contain all these members.)

Member	Description
AEPL	Assembler macro that maps all fields passed to an assembler program by JOB/SCAN.
ASMP	Sample assembler program used with JOB/SCAN Standards Facility. This sample includes site-specific logic.
CCHK	COBOL copy member to validate the CEPL member.
CEPL	COBOL copy member that contains all data definitions that interface the standards program with JOB/SCAN.
CEECMPLK	Compile and link ASG-JOB/SCAN standards programs using enterprise COBOL compiler for z/OS.
CERR	COBOL copy member that contains a table-handling procedure for storing standards program error messages created as the current statement is validated.
CKIT	Sample Enterprise COBOL program shell for the Standards Facility. Contains code necessary to access the JCL at any level.
CSMP	Sample Enterprise COBOL program used with JOB/SCAN Standards Facility. This is an example of how site-specific logic can be added to the CKIT shell.
CTRKIT	A starter kit in COBOL used to perform exercises in the training program. Also used by Standards Prompter.
C2SMP	Sample Enterprise COBOL program used with JOB/SCAN Standards Facility. This is an example of how site-specific logic can be added to the C2KIT shell.
EPLDOC	Member containing detailed definitions of all fields in Standards Facility Extended Parameter List (CEPL). Designed for online browsing and assists in writing of standards programs. JCL in member EPLPRNT can be used to print hardcopy.
EPLPRNT	Member containing executable JCL to print member EPLDOC.
JSVERIFY	Member containing JCL to run Standards Facility field display program to assist with debugging a standards program.
JSXTDG0	COBOL standards program that checks GDG-ALL datasets for generations which might have been migrated. Calls assembler subroutine JSXTDG1.
JSXTDG1	Assembler subroutine that performs actual validation of GDG-ALL datasets for generations that might have migrated. Called by COBOL program JSXTDG0 must run in TSO environment because invokes TSO Service facility.

Member	Description
JSXTDJES	Sample COBOL program which generates SJJSPARM library member JSOPTJS. Contains a JIIPA table built from DD statements found in PROC00 in JCL member being parsed. Program Product Interface (PPI) RUN OPTION "JES2PROC" enables the SJJSPARM library member.
RKIT	Sample REXX program shell for the Standards Facility. Contains code necessary to access the JCL at any level.
RSMP	Sample REXX program used with JOB/SCAN Standards Facility. This is an example of how site-specific logic can be added to the RKIT shell.

## SJJSDEMO Library

The SJJSDEMO library contains job streams for producing sample JOB/SCAN reports. These members are referenced in the *JOB/SCAN Structured Learning Program (SLP)* examples. Additionally, there are sample JCL Change and JCL Reformatter Rules.

Sample JCL is validated by specifying SJJSDEMO as the input library on the JCL Validation panel. For an explanation of how to validate sample JCL, see the [User Information - JOB/SCAN](#).

This table describes the members in the SJJSDEMO library.

Member	Description
CHGJCL	JCL which is the object of sample JCL Change Rules found in members CHGRULE1-CHGRULE6.
CHGRULE1-6	Sample JCL Change Rule members for JCL found in member CHGJCL.
CHNGJCL	Another JCL member which is the object of sample JCL Change Rules found in members CHNGRULE1-CHNGRULE6.
CHNGJCL1-4	Another set of JCL Change Rules for JCL found in member CHNJCL.
FORMJCL	Sample JCL used to demonstrate the JCL Reformatter function.
FORMJCL2	Another JCL sample used to demonstrate the JCL Reformatter function.
PJDB0010	Sample JCL used to demonstrate DB2 control statement JCL Validation.
PJGL0020-56	Sample JCL used to demonstrate the JCL Validation function.
PJPR0020-56	Additional JCL samples used to demonstrate the JCL Validation function.

Member	Description
PROC1	Sample PROC used to demonstrate the JCL Validation function.
SAMPLJCL	Sample JCL used to demonstrate the JCL Validation function.
STNDJCL	Sample JCL used to demonstrate the JCL Standards function.

---

# Product Usage Facility

JOB/SCAN's Product Usage Facility tracks the use of JOB/SCAN at the site-, machine-, and user- levels. The Facility tracks details about:

- numbers of lines and members scanned
- JCL and standards errors found.

Administration of this facility includes:

- maintenance of the VSAM data file
- activation/deactivation of users
- starting and stopping collection periods

## Collection Periods

The Product Usage Facility in JOB/SCAN employs the concept of *collection periods* ("period"). Only one period is actively collecting data at any time. The first period for existing customers is period 00000 containing all the data from releases prior to 6.2.6A. JOB/SCAN administrators start and stop collection periods at suitable intervals (weekly, quarterly, yearly, and so on). See [Managing Collection Periods](#).

The period is the basic unit for managing product usage data.

## Viewing Usage Data

Usage data is readily available for viewing from the ASG-JOB/SCAN Primary Menu. See [Displaying Site-wide Product Usage Information](#), [Displaying Product Usage by Period](#), and [Displaying Usage Data by Individual](#).

## Exporting and Transferring Usage Data

The Product Usage Facility in JOB/SCAN contains an export utility to export usage data to an intermediate file, which is then transferred to a target machine. This utility writes the data in a comma-delimited format. Data export is a one-way transfer: JOB/SCAN does not import usage data. See [Exporting Usage Data](#).

- [Creating and Maintaining a Product Usage File](#)
- [Activating and the Product Usage Facility](#)
- [Managing Collection Periods](#)
- [Displaying Site-wide Product Usage Information](#)
- [Displaying Product Usage by Period](#)
- [Displaying Usage Data by Individual](#)
- [Managing Users](#)
- [Exporting Usage Data](#)

## Creating and Maintaining a Product Usage File

When JOB/SCAN is installed, its Product Usage Facility is set up to write data automatically to a VSAM file. See [Step 7—Set Up the Product Usage Facility](#) for more information about creating the VSAM file.

Follow the usual site procedures that apply to production VSAM files for:

- periodically making backups
- reorganizing
- checking file allocation space. The Product Usage Facility VSAM file continues to grow because records are added for each new JOB/SCAN use. Check the VSAM file allocation space on a regular basis.

Some Usage file maintenance functions can be performed by the utility J00YPUMT. Specifically, collection periods can be created at intervals determined by the product administrator, and deleted upon request. The JCL shown in the figure can be used to maintain a Product usage file:

### SJSSAMP library member PMAINT

```
//PMAINT JOB (ACCT), 'PROGRAMMER NAME'
//*****
//* THIS IS A SAMPLE JOB TO CREATE OR DELETE PRODUCT USAGE
//* FILE COLLECTION PERIODS.
//*
//* PARAMETERS FOR J00YPUMT ARE:
//*      PARM='CREATE'      TO CREATE A NEW COLLECTION PERIOD
//*      PARM='DELETE=NN'  TO DELETE A COLLECTION PERIOD.
//*                          NN IS A PERIOD NUMBER TO DELETE.
//*
//* REF:  SAMPLIB (PMAINT)
//*
//* NOTE: MODIFY JOB STATEMENT AND ITEMS INDICATED BY ARROWS <==
//*****
//STEP01 EXEC PGM=J00YPUMT, PARM=          <= CREATE OR DELETE=NN
//STEPLIB DD  DISP=SHR, DSN=ASG.SJJSIMOD  <= YOUR ASG LOAD LIBRARY
//JLLOD  DD  DISP=SHR, DSN=ASG.SJJSIMOD  <= YOUR ASG LOAD LIBRARY
//JLLPUF DD  DISP=SHR,
//      DSN=ASG.PRODUCT.USAGE          <== PRODUCT USAGE FILE
//JLSSR  DD  SYSOUT=*
//SYSPRINT DD SYSOUT=*, DCB=(LRECL=133, BLKSIZE=1330)
***** Bottom of Data *****
```

## Activating and the Product Usage Facility

Activating the gathering of JOB/SCAN usage statistic requires two detail statements attached to the ACCESS CONTROL statement. See [JOB/SCAN Usage File Detail Statement](#) and [JOB/SCAN Usage Report Detail Statement](#) for full details.

### To activate the JOB/SCAN Product Usage data gathering for all users

1. Edit member JSOPT01 in the SJJSPARM library.
2. Place the following detail statements in the Access Control statement:

```
JOB/SCAN USAGE FILE=YES
JOB/SCAN USAGE REPORT=MSG=ADVISORY
```

**JSOPT01 statements activating the Product Usage Facility**

```

** JSOPT01  ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** *
** THIS MEMBER ENABLES ACCESS TO JOB/SCAN AND **
** CONTROLS USER ACCESS TO PANELS AND EXTENDED FUNCTIONS. **
** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** *
...
ACCESS CONTROL
  INITIALIZATION      =ALL
  JOB/SCAN            =ALL, STANDARDS=ANY, PARM=
  SELECT LISTS       =ALL
  OPTIONS             =USERID=
  ADMINISTRATOR       =USERID=
  JOB/SCAN USAGE FILE=YES
  JOB/SCAN USAGE REPORT=MSG=ADVISORY
...

```

- Put this File Allocation statement in JSOPT02 or its equivalent so JOB/SCAN will write usage data to the correct VSAM file (allocated during installation):

```
JLLPUF=SHR, DSN=datasetname
```

**JSOPT02 statements activating the Product Usage Facility**

```

...
FILE ALLOCATIONS
  JLLPUF  =DSN=your.VSAM.dsn.here,SHR  VSAM dataset alloc'd at install
...

```

**Activating and Deactivating Userid Encoding for JOB/SCAN Usage Reports**

On several panels and in several reports, JOB/SCAN includes the userids of the users who validated JCL. JOB/SCAN can encode the userids to preserve user anonymity.

**To activate userid encoding in the JOB/SCAN Product Usage facility**

- Start the JOB/SCAN Usage Report from the ASG-JOB/SCAN Primary Menu. JOB/SCAN displays the Product Usage Administration panel.
- Select Enable User ID Encoding. (Only JOB/SCAN administrators see this option.) JOB/SCAN displays confirmation dialog.
- Press Enter to start encoding; press Cancel or Exit to keep userids.
- JOB/SCAN displays Encoding Enabled, in the top-right message area of the Product Usage Administration panel.

**To deactivate userid encoding**

- Start the JOB/SCAN Usage Report from the ASG-JOB/SCAN Primary Menu. JOB/SCAN displays the Product Usage Administration panel.
- Select Disable User ID Encoding.
- JOB/SCAN displays Encoding Disabled, in the top-right message area of the Product Usage Administration panel.

## Prohibiting Writing Statistics to Product Usage Report File

### To keep JOB/SCAN from writing usage statistics to the Usage report

1. Edit member JSOPT01 in the SJJSPARM library.
2. Place the following detail statement in the Access Control statement:

```
JOB/SCAN USAGE FILE =NO
```

Make sure that all other JOB/SCAN Usage Report File detail statements are commented out or deleted from the member.

## Managing Collection Periods

A collection period in JOB/SCAN's Product Usage Facility is a user-established time period during which JOB/SCAN collects data. Product administrators start, stop and delete periods using items on the PRODUCT USAGE ADMINISTRATION menu. For each use, JOB/SCAN computes statistics for each period by user and keeps a running master total for the site for all periods.

### Viewing the List of Collection Periods

#### To view collection periods

- Select View Collection Periods from the Product Usage Administration menu to view a list of periods.

#### Viewing the collection periods

```

----- PRODUCT USAGE ADMINISTRATION -----
SELECTION ==> 2

1 - View Site Usage History
2 - View Collection Periods
3 - Set New Data Collection Period
4 - List Inactive Users
5 - Prepare Usage Data For transfer
X - Exit

USERID - ASGUSR
PREFIX - ASGUSR
DATE -
APPLID - JS

```

JOB/SCAN displays the JOB/SCAN Period Summary.

#### List of Collection Periods

```

----- JOB/SCAN PERIOD SUMMARY ----- Row 1 to 2 of 2
COMMAND ==> SCROLL ==> PAGE

Enter S to view or D to delete
----- USAGE PERIODS -----
Usage Period
Cmd  Period  First Use Date  Last use Date  Created by  Created on
-    00001    2005-04-19     2005-04-19    ASGUSR     2005-04-19
-    00000    1995-01-26     2005-04-06    ASGUSR     2005-04-19
***** Bottom of Data *****

```

### Commands on Periods

In the Cmd column for each period, use:

<b>S</b>	Select period for viewing.
<b>D</b>	Delete period (but period 00000 cannot be deleted).

**Fields**

The bottom portion of the panel contains these fields:

Panel Field	Description
PERIOD	The JOB/SCAN-assigned number for a time period of JOB/SCAN use.
FIRST-USE	The first date of JOB/SCAN use for the period.
LAST-USE	The last date of JOB/SCAN use for the period.
CREATED BY	The userid of the person who created the period.
CREATED ON	The date the period was created.

**To delete a period**

- On the JOB/SCAN Period Summary Panel, use the **D** command to delete a period

**List of Collection Periods**

```

----- JOB/SCAN PERIOD SUMMARY ----- Row 1 to 2 of 2
COMMAND ==>                               SCROLL ==> PAGE
Enter S to view or D to delete
----- USAGE PERIODS -----
Cmd   Period   First Use Date   Last use Date   Created by   Created on
D     00003    2005-04-19      2005-04-19     ASGUSR      2005-04-19
      00000    1995-01-26      2005-04-06     ASGUSR      2005-04-19
-----
***** Bottom of Data *****
    
```

JOB/SCAN displays this confirmation popup:

**Confirming a period deletion**

```

JOB/SCAN 7.7.0 - Press ENTER to continue
----- Confirm Action -----
DELETE PERIOD: 00003
Press ENTER to confirm.
Press CANCEL or EXIT to cancel.
    
```



## Starting a New Usage Period

**To close the current period and start a new one**

1. Select Set New Data Collection Period.

### Viewing the collection periods

```

----- PRODUCT USAGE ADMINISTRATION -----
SELECTION ==> 3
1 - View Site Usage History
2 - View Collection Periods
3 - Set New Data Collection Period
4 - List Inactive Users
5 - Prepare Usage Data For transfer
X - Exit
USERID - ASGUSR
PREFIX - ASGUSR
DATE -
APPLID - JS
    
```

JOB/SCAN displays the Confirm Action panel.

2. Press Enter to start a new collection period.

### Confirming the closing of a period

```

Confirm Action
CREATE NEW COLLECTION PERIOD?
Press ENTER to confirm.
Press CANCEL or EXIT to cancel.
    
```

JOB/SCAN displays the PRODUCT Usage Administration menu and posts a message to the message area of the panel.

### Message confirming the start of a new period

```

D00PUADM ----- PRODUCT USAGE ADMI          NEW PERIOD STARTED
SELECTION ==>
1 - View Site Usage History
2 - View Collection Periods
3 - Set New Data Collection Period
4 - List Inactive Users
5 - Prepare Usage Data For transfer
X - Exit
USERID - ASGUSR
PREFIX - ASGUSR
DATE - 2005-04-19
APPLID - JS
    
```

3. Re-select View Collection Periods to show the new period.

Note the blank First Use Date field and the NO J/S RUN entry in the Last use Date field.

### New Period summary showing newly created, unused period

```

----- JOB/SCAN PERIOD SUMMARY ----- Row 1 to 3 of 3
COMMAND ==>
Enter S to view or D to delete
----- USAGE PERIODS -----
SCROLL ==> PAGE
    
```

Cmd	Period	First Use Date	Last use Date	Created by	Created on
-	00002		NO J/S RUN	ASGUSR	2005-04-19
-	00001	2005-04-19	2005-04-19	ASGUSR	2005-04-19
-	00000	1995-01-26	2005-04-06	ASGUSR	2005-04-19

## Displaying Site-wide Product Usage Information

To display the information in the *JOB/SCAN Product Usage VSAM file*

- Choose Site Usage History from the PRODUCT Usage Administration menu.

### Selecting Usage History

```

----- PRODUCT USAGE ADMINISTRATION -----
SELECTION ==> 1

1 - View Site Usage History
2 - View Collection Periods
3 - Set New Data Collection Period
4 - List Inactive Users
5 - Prepare Usage Data For transfer
X - Exit

USERID - ASGUSR
PREFIX - ASGUSR
DATE - 2005-05-27
APPLID - JS

```

JOB/SCAN displays the JOB/SCAN Usage Analysis panel containing totals for all periods, including deleted periods:

### JOB/SCAN Site Usage History Panel

```

----- SITE USAGE HISTORY -----
COMMAND ==>

First Used: 2003-05-08 Release: *      Items Scanned  Use Count
Last Used: 2005-05-27 Release: 7.7.0  Lines: 1116643  ISPF : 3674
Errors Reported
  JCL      : 144428  Procs: 17154   Edit : 11091
  Standards : 15834  Steps: 59952  Total: 18700
                      Dds : 333476

```

This table lists and describes fields in the Site Usage History panel:

Panel Field	Description
FIRST USED	Date the tool was first used at site.
FIRST USED RELEASE	The release of JOB/SCAN first used.
LAST USED	Date the tool was most recently used.
LAST USED RELEASE	The release of JOB/SCAN most recently used.
ERRORS REPORTED: JCL	Total number of JCL errors found to date.
ERRORS REPORTED: STANDARDS	Total number of standards violations found to date.
LINES	Total number of JCL lines scanned to date.

Panel Field	Description
JOBS	Total number of JCL jobs scanned to date.
PROCS	Total number of procs scanned to date.
STEPS	Total number of steps scanned to date.
DDS	Total number of DDS from the jobs scanned to date.
USE COUNT: ISPF	Total number of ISPF panel validations.
USE COUNT: BATCH	Total number of batch validations.
USE COUNT: EDIT	Total number Edit macro validations.
USE COUNT: TOTAL	Total number of panel + batch + Edit macro validations.

## Displaying Product Usage by Period

### To display product usage by period

1. Select View Collection Periods from the Product Usage Administration menu.

#### Initiating viewing periods

```

----- PRODUCT USAGE ADMINISTRATION -----
SELECTION ==> 2

1 - View Site Usage History
2 - View Collection Periods
3 - Set New Data Collection Period
4 - List Inactive Users
5 - Prepare Usage Data For transfer
X - Exit

USERID - ASGUSR
PREFIX - ASGUSR
DATE - 2005-05-27
APPLID - JS
    
```

2. On the JOB/SCAN Period Summary panel, use the S command to view the detailed usage for a period.

#### Period usage summary

```

----- JOB/SCAN PERIOD SUMMARY ----- Row 1 to 2 of 2
COMMAND ==>
Enter S to view or D to delete
----- USAGE PERIODS -----
Usage Period
Cmd  Period  First Use Date  Last use Date  Created by  Created on
S    00003   2005-05-27     2005-05-27   ASGUSR     2005-05-27
     00002   2005-05-27     2005-05-27   ASGUSR     2005-05-25
-----
***** Bottom of Data *****
    
```

3. JOB/SCAN displays the Usage for Selected Period panel.

**Period usage detail**

```

----- USAGE FOR SELECTED PERIOD ----- Row 1 to 3 of 3
COMMAND ==>                               SCROLL ==> PAGE
Period: 00003          Created on: 2005-05-27
Release   : 7.7.0
Created by: ASGUSR
First Use: 2005-05-27  Release: 7.7.0      Lines:      10   ISPF :      2
Last Use: 2005-05-27  Release: 7.7.0      Jobs :       2   Batch:      0
Errors Reported      Procs:       0   Edit :      0
      JCL   :           0           Steps:      2   Total:      2
      Standards :       0           Dds :       2
----- Active Users -----
Use SORT a b in COMMAND line. a = U/F/L/T/J  b = A/D
Enter S to select I to inactivate one or more users

Cmd   Userid      First-Use      Last-Use      Total-Uses    JCL Errors
--   -
--   ASGUSR      2005-05-27 07:22 2005-05-27 07:32      2           0
--   ASGUSR1     2005-05-27 07:22 2005-05-27 07:32      2           0
--   ASGUSR2     2005-05-27 07:22 2005-05-27 07:32      2           0

```

**Panel Commands**

Use the SORT command to sort the table on any of the fields in this table.

For this field	Use this parameter
USERID	U
FIRST-USE	F
LAST-USE	L
TOTAL-USES	T
JCL ERRORS	J

Use SORT with A or D sub-params to sort in ascending or descending order. This example sorts the panel by userid in ascending order:

```
SORT U A
```

**Line Commands**

In the Cmd field for each Userid use:

**S** Select Userid for viewing. See [Displaying Usage Data by Individual](#).

**I** Inactivate the user. See [Inactivating a User](#).

**Fields**

This table lists the fields of the Period Usage panel.

Panel Field	Description
PERIOD	The numerical id of the period.
CREATED ON	The date the period was created.
RELEASE	The JOBSCAN release used to create this period.
CREATED BY	The user id that created the period.
FIRST USED	Date the tool was first used at site.
LAST USED	Date the tool was most recently used.
ERRORS REPORTED: JCL	Total number of JCL errors found to date.
ERRORS REPORTED: STANDARDS	Total number of standards violations found to date.
LINES	Total number of lines of JCL scanned to date.
JOBS	Total number of JCL jobs scanned to date.
PROCS	Total number of procs scanned to date.
STEPS	Total number of steps scanned to date.
DDS	Total number of DDS referenced in the jobs scanned to date.
USE COUNT:	Total number of times JOB/SCAN was used.
USE COUNT: ISPF	Total number of times JOB/SCAN used from ISPF panels.
USE COUNT: BATCH	Total number of times JOB/SCAN used in batch mode.
USE COUNT: EDIT	Total number of times JOB/SCAN used from Edit macros.
USERID	The user IDs that have used JOB/SCAN during this period
FIRST-USE	The date the user first used JOB/SCAN
LAST-USE	The date the user most recently used JOB/SCAN
TOTAL-USES	The number of times the user ran JOB/SCAN
JCL ERRORS	The total number of errors found during this period

## Displaying Usage Data by Individual

JOB/SCAN computes usage information for each use by an individual userid.



In its reports, JOB/SCAN includes the userids for users who validate, reformat, or change JCL. However, if the actual userids are not required, see [Activating and Deactivating Userid Encoding for JOB/SCAN Usage Reports](#) to turn on userid encoding. The remaining sample panels show userids unencoded; when encoding is turned on, the Userid fields contain an encoded strings.

### To see this data for any userid

1. Type `s` in the Cmd field next to the Userid in the Usage For Selected Period panel.

#### Selecting a Userid for to view usage data

```

----- USAGE FOR SELECTED PERIOD ----- Row 1 to 3 of 3
COMMAND ==>                               SCROLL ==> PAGE
Period: 00003          Created on: 2005-05-27
Release   : 7.7.0
Created by: ASGUSR
First Use: 2005-06-20  Release: 7.7.0      Lines:      3549  ISPF :    250
Last Use: 2005-06-20  Release: 7.7.0      Jobs :       261  Batch:     3
Errors Reported                               Procs:       16  Edit :     0
  JCL      :      152                          Steps:      288  Total:    253
  Standards :    2829                          DDs :      1210
----- Active Users -----
Use SORT a b in COMMAND line. a = U/F/L/T/J  b = A/D
Enter S to select, I to inactivate one or more users

Cmd   Userid      First-Use      Last-Use      Total-Uses    JCL Errors
S     ASGUSR      2005-06-20 14:45 2005-06-20 15:02      245         83
-     ASGUSR1     2005-06-20 14:46 2005-06-20 15:11         5          55
-     ASGUSR2     2005-06-20 15:02 2005-06-20 15:03         3          14

```

JOB/SCAN displays the Usage For userid -Period nnnnn panel.



For user ASGUSR, who has amassed 245 uses of JOB/SCAN (see above figure), all of the uses are summarized in the above figure. However, in the details of the below figure, only the latest 240 uses are available for review.

2. Select an instance of use from the Usage for userid panel with `S` command.

**Selecting an instance from the Usage summary for an individual user**

```

----- USAGE FOR ASGUSR - PERIOD 00002 ----- Row 1 to 3 of 240
COMMAND ==>                                SCROLL ==> PAGE

User:  ASGUSR

First Use: 2005-06-20 14:45 Release: 7.7.0   Items Scanned Usage Count
Last Use:  2005-06-20 15:02 Release: 7.7.0   Lines:    2695 ISPF :   245
Errors Reported Procs:      0 Edit :      0
          JCL      :    83 Steps:    245 Total:   245
          Standards :    60 Dds :    980

Enter S to select                               Standards
Cmd   Date           Type  JCL errors  Msgs  Exit Name  Release
  S   2005-06-20 15:02 ISPF      0      0   *NONE*   6.27A0
  _   2005-06-20 15:02 ISPF      0      0   *NONE*   6.27A0
  _   2005-06-20 15:02 ISPF      0      0   *NONE*   6.27A0
...
    
```

JOB/SCAN displays the details panel for that usage record.

**Usage details for an individual user**

```

----- USAGE DETAILS FOR ASGUSR -----
COMMAND ==>

User:  ASGUSR           Date:    2005-05-25 15:46
          Release:    7.7.0
          Type:      ISPF

          Items Scanned           Errors Reported
          Jobs :      1             JCL      :      0
          Procs:      0             Standards :      0
          Steps:      1
          Dds :      1
          Lines:      5             Standards Exit: *NONE*
    
```

**Fields**

This table lists the fields for the Usage for userid and Usage Details for userid panels.

Panel Field	Description
FIRST-USE	The date the user first used JOB/SCAN
LAST-USE	The date the user most recently used JOB/SCAN
USE COUNT:	Total number of times JOB/SCAN was used.
USE COUNT: ISPF	Total number of times JOB/SCAN used from ISPF panels.
USE COUNT: BATCH	Total number of times JOB/SCAN used in batch mode.
USE COUNT: EDIT	Total number of times JOB/SCAN used from Edit macros.
DATE	Date and time of each run.

Panel Field	Description
RELEASE	The release of JOB/SCAN that produced the data.
TYPE	Run type: ISPF; EDIT; BATCH.
JOBS	Number of jobs in the run.
PROCS	Number of procs in the run.
STEPS	Number of steps in the run.
DDS	The number of DDS in the run.
LINES	Total number of lines of JCL scanned to date.
ERRORS: JCL	Number of JCL errors found.
ERRORS: STANDARDS	Number of standards violations found.
STANDARDS EXIT	The name of the standards exit used. *NONE* means that no exit was used.

## Managing Users

Managing users in the Product Usage Facility entails inactivating users to remove them from the Product Usage Facility reports. Their usage history remains in the file but is no included in the reports. An administrator may at any time reactivate users to view their data. Individually inactivated uses may cause their own reactivation at any time they use the validation function of JOB/SCAN.

### Inactivating a User

The inactive status only excludes a user's data from the Product Usage reports; an inactive user may still access and use JOB/SCAN at any time (which reactivates the userid for reporting purposes).

#### *To inactivate a user*

1. Type the `␣` line command to select a userID.
2. Press Enter.



**Initiating the inactivation of a user**

```

D00PUSP ----- USAGE FOR SELECTED PERIOD ----- Row 1 to 26 of 71
COMMAND ==>                                     SCROLL ==> PAGE
Period: 00000          Created on: 2005-04-19
Release   : 7.7.0
Created by: ASGUSR
First Use: 1995-01-26  Release: 7.7.0          Lines: 99068929  ISPF :   8649
Last Use: 2005-04-06  Release: 7.7.0          Jobs  :   2547288  Batch: 259102
Errors Reported                               Procs: 2382016   Edit :   16763
      JCL      :   6370704                      Steps: 9010155   Total: 284514
      Standards : 2965286                      Dds   : 17391935
----- Active Users -----
Use SORT * in COMMAND line. * = U/F/L/T/J
Enter S to select, I to inactivate one or more users.

Cmd   Userid      First-Use      Last-Use      Total-Uses    JCL Errors
---   -
I     ASGUSR     1999-04-28 00:50 1999-04-28 00:50      1           0
...   ASGUSR     2004-12-09 10:12 2004-12-09 10:12      1           0
...

```

3. JOB/SCAN displays the Confirm Action dialog. Press Enter.

**Confirming the inactivation of a user**

```

----- Confirm Action -----
INACTIVATE ASGUSR
Press ENTER to confirm.
Press CANCEL or EXIT to cancel.

```

**Listing Inactivated Users**

**To view a list of inactivated users**

- Select List Inactive Users on the Product Usage Administration panel.

**Listing the inactive users**

```

----- PRODUCT USAGE ADMINISTRATION -----
SELECTION ==> 4

1 - View Site Usage History          USERID - ASGUSR
2 - View Collection Periods          PREFIX  - ASGUSR
3 - Set New Data Collection Period    DATE   - 2005-05-25
4 - List Inactive Users              APPLID - JS
5 - Prepare Usage Data For transfer
X - Exit

```

JOB/SCAN displays the Inactivated Users panel:

**List of inactive users**

```

----- INACTIVATED USERS ----- Row 1 to 1 of 1
COMMAND ==> SCROLL ==> PAGE

Enter S/A/R to reactivate user(s).

Cmd  Userid      Period      Inactivated On      Inactivated By
   R  ASGUSR      00002      2005-04-19         ASGUSR
***** Bottom of Data *****

```

**Reactivating Users****To reactivate a user**

1. View the list of inactive users.
2. Use the R command.

**List of inactive users**

```

D00PUDLU ----- INACTIVATED USERS ----- Row 1 to 1 of 1
COMMAND ==> SCROLL ==> PAGE

Enter S/A/R to reactivate user(s).

Cmd  Userid      Period      Inactivated On      Inactivated By
   R  ASGUSR      00002      2005-04-19         ASGADM
***** Bottom of Data *****

```

3. JOB/SCAN displays this popup. Press ENTER to confirm.

**Reactivating a user: confirmation pop-up**

```

REACTIVATE USER ASGUSR

Press ENTER to confirm.

Press CANCEL or EXIT to cancel.
F1=HELP      F2=SPLIT     F3=END
F4=RETURN    F5=RFIND     F6=RCHANGE

```

**Exporting Usage Data**

JOB/SCAN provides an export utility to give customers the choice of viewing their JOB/SCAN usage data with other tools. Any program that can accept comma-delimited data, such as spreadsheets and some word processing programs, will be able to organize the data from the exported file.



The raw, exported data file is not meant to be human readable; first transfer it to a target machine and then load it into a spreadsheet or other suitable program.

**Preparing the Export File through Panels**

See topic - [Step 7—Set Up the Product Usage Facility](#) for instructions about how to allocate the JLLPUDLT DD to the

dataset.

### To prepare the data export file

1. From the Product Usage Administration panel, choose Prepare Usage Data for Transfer.

#### Selecting Export Facility

```

----- PRODUCT USAGE ADMINISTRATION -----
SELECTION ==> 5

1 - View Site Usage History
2 - View Collection Periods
3 - Set New Data Collection Period
4 - List Inactive Users
5 - Prepare Usage Data For transfer
X - Exit

USERID - ASGUSR
PREFIX - ASGUSR
DATE - 2005-05-26
APPLID - JS

```

2. JOB/SCAN displays the PREPARE USAGE DATA panel. You may select any number of periods with the S line command to be included in the report.

#### Selecting a period to export

```

----- JOB/SCAN - Select Usage Data ----- Row 1 to 3 of 3
COMMAND ==>                                SCROLL ==> PAGE

----- PREPARE USAGE DATA -----

Prepare product usage data for download.
Select one or more periods below, then press ENTER to convert
usage data to down load format. This data will be written to
the DD name JLLPUDLT.

Cmd  Status  Period Release  First Use  Last use  By  On
S    CURRENT  00002  7.7.0    2005-05-25 2005-05-25 ASGUSR 2005-05-25
S    DELETED  00001  7.7.0    2005-05-25 2005-05-25 ASGUSR 2005-05-25
-    -        00000  *        2003-05-08 2005-05-25 ASGUSR 2005-05-25

```

3. Select one or more periods and press Enter; then press End. When the export finishes, JOB/SCAN displays the FILE PREPARED message:

#### Portion of the Select Usage Data panel indicating data export is successful

```

----- JOB/SCAN - Select Usage Dat----- FILE PREPARED
COMMAND ==>                                SCROLL ==> PAGE

...

```

Any period listed may be exported. For subsequent exportations of data, JOB/SCAN displays a message warning that previously exported data will be overwritten.



Exporting a deleted file will lead to an exported file containing only header data.

### Preparing the Export File through Batch Jobs

Usage data can also be exported through batch jobs. The sample JCL in the figure (SJSSAMP library member PFORMAT)

can be used.

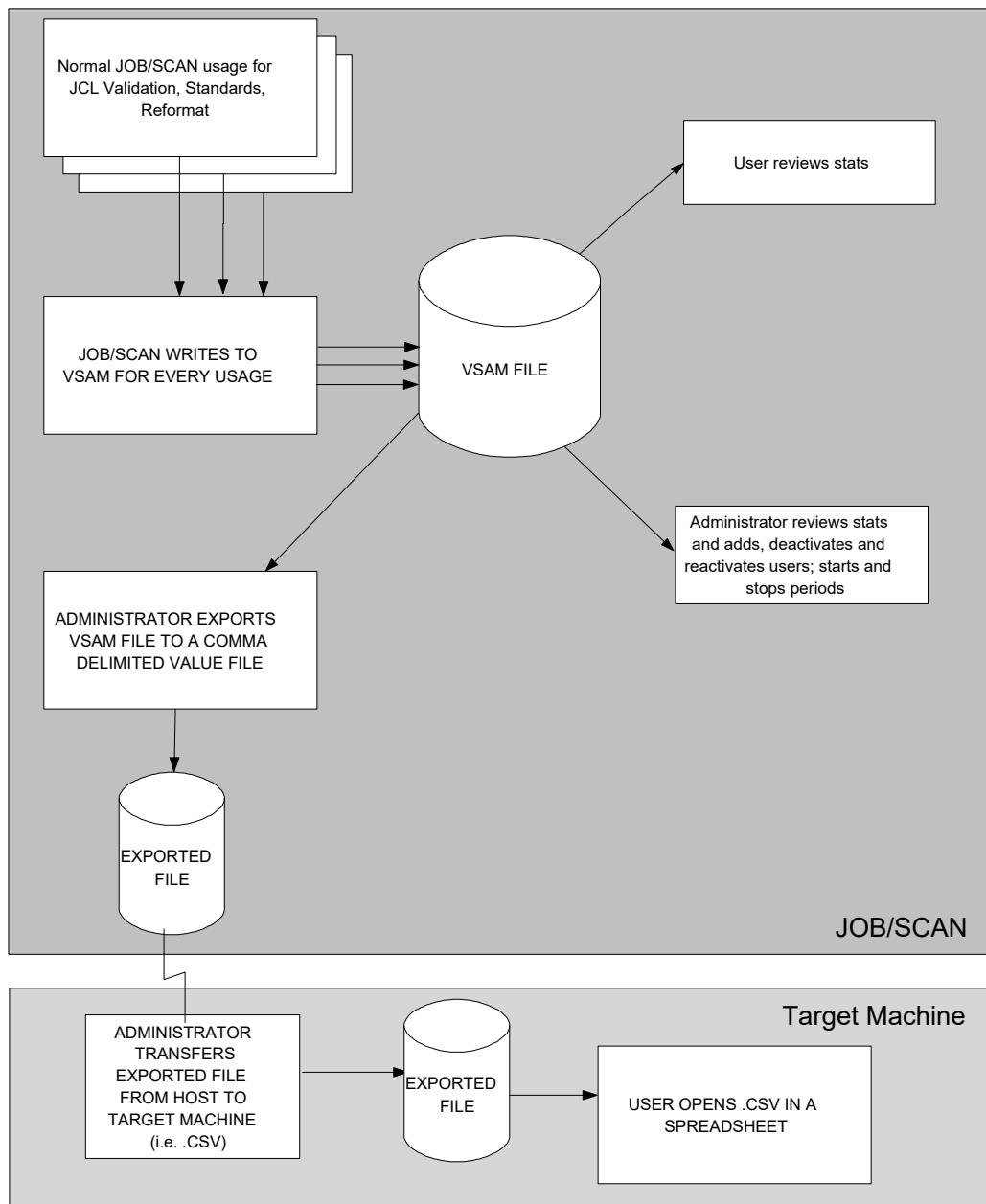
#### Batch JCL from SJSSAMP library member PFORMAT to export usage information

```
//PFORMAT JOB (ACCT),'PROGRAMMER NAME'
//*****
//* THIS IS A SAMPLE JOB TO FORMAT ONE OR MORE COLLECTION PERIODS *
//* FROM A JOB/SCAN PRODUCT USAGE FILE. *
//* *
//* REF: SAMPLIB(PFORMAT) *
//* *
//* NOTE: MODIFY JOB STATEMENT AND ITEMS INDICATED BY ARROWS <== *
//*****
//STEP01 EXEC PGM=J00YPUDN,
//          PARM='01,02,03...' <== ONE OR MORE PERIOD NUMBERS
//STEPLIB DD DISP=SHR,DSN=ASG.SJJSLSMOD <== YOUR ASG LOAD LIBRARY
//JLLOD DD DISP=SHR,DSN=ASG.SJJSLSMOD <== YOUR ASG LOAD LIBRARY
//*
//JLLPUF DD DISP=SHR,
//          DSN=ASG.PRODUCT.USAGE <== PRODUCT USAGE FILE (IN)
//JLLPUDLT DD DISP=SHR, <== FORMATED USAGE FILE (OUT)
//          DSN=DSSE.USAGE.EXPORT
//JLSSR DD SYSOUT=*
***** Bottom of Data *****
```

#### Downloading the Usage Data

Use any convenient means to download the usage file to the target machine.

### Transferring Usage Data From JOB/SCAN



### Organization of the Export file

The Export file is a sequential dataset. It contains several record types:

- one site-header record followed by the site-data record
- one or more period-header records each followed by one period-data record for each period
- each period contains one user-header record followed by one or more userID-data records.

The figure shows the export file when viewed in a text-only view.

Sample text-only view of the downloaded export file.

```

**SITE-RECORD                ,SIT-CR-U,SIT-CR-DT                ,SIT-PC-U,SIT-PC-DT
JOBS,TOT-PRCS,TOT-STEP,TOT-DD ,TOT-LINE,TOT-JCLE,TOT-STDE
AAA_SITE_MASTER              ,SYSUSR ,2005-06-09 07:06,SYSUSR ,2006-04-12
,00032,00018338,00009934,00318107,03224995,02993261,10648862,20468452,21723089
**PERIOD-RECORD              ,PER-STAT,PER-CR-U,PER-CR-DT                ,PER-CR-REL
ISPF,TOT-BAT ,TOT-JOBS,TOT-PRCS,TOT-STEP,TOT-DD ,TOT-LINE,TOT-JCLE,TOT-STDE
A-PERIOD00032ZZZZZZZMASTER  ,CURRENT ,SYSUSR ,2006-04-12 08:54,6.29A0
,00000092,00000183,00007684,00085291,00078732,00221642,00434133,03040566,00189
**USER                        ,USR-FA-DT                ,USR-FA-REL,USR-LA-DT                ,USR-LA-REL,USR-STAT
SYSUSR 00032,2006-04-18 09:00,6.2.8A ,2006-05-15 14:17,6.2.8B ,ACTIVE
**DETAIL, PERI ,DATE                ,RLSE                ,TYPE ,EXIT                ,JOBS ,PROCS ,
SYSUSR ,00032,2006-05-15 14:17,6.2.8B ,EDIT ,*NONE* ,00000001,00000000,
SYSUSR ,00032,2006-04-20 12:48,6.2.8A ,EDIT ,*NONE* ,00000001,00000000,
SYSUSR ,00032,2006-04-20 12:47,6.2.8A ,ISPF ,*NONE* ,00000001,00000000,
SYSUSR ,00032,2006-04-20 12:47,6.2.8A ,ISPF ,*NONE* ,00000001,00000000,
SYSUSR ,00032,2006-04-20 12:47,6.2.8A ,ISPF ,*NONE* ,00000001,00000000,
SYSUSR ,00032,2006-04-18 09:00,6.2.8A ,EDIT ,*NONE* ,00000001,00000000,
**USER                        ,USR-FA-DT                ,USR-FA-REL,USR-LA-DT                ,USR-LA-REL,USR-STAT
    
```

The below figure shows a word-processing table formatting the data from the above figure.

Sample word-processing table showing the transferred usage file

**SITE-RECORD	SIT-CR-U	SIT-CR-DT	SIT-PC-U	SIT-PC-DT
AAA_SITE_MASTER	SYSUSR	2005-06-09 07:06	SYSUSR	2006-04- 12 08:54
**PERIOD-RECORD	PER-STAT	PER-CR-U	PER-CR- DT	PER-CR- REL
A- PERIOD00032ZZZZZZZMASTER	CURRENT	SYSUSR	2006-04- 12 08:54	6.29A0
**USER	USR-FA- DT	USR-FA-REL	USR-IA- DT	USR-LA- REL
SYSUSR 00032	2006-04- 18 09:00	6.2.8A	2006-05- 15 14:17	6.2.8B
**DETAIL	PERI	DATE	RLSE	TYPE
SYSUSR	00032	2006-05-15 14:17	6.2.8B	EDIT
SYSUSR	00032	2006-04-20 12:48	6.2.8A	EDIT
SYSUSR	00032	2006-04-20 12:47	6.2.8A	ISPF
SYSUSR	00032	2006-04-20 12:47	6.2.8A	ISPF
SYSUSR	00032	2006-04-20 12:47	6.2.8A	ISPF

## Export File Fields

### Site Fields

Field	Description
SIT-CR-U	User ID that created the first period.
SIT-CR-DT	Date Period created.
SIT-PC-U	User ID that created Current period.
SIT-PC-DT	Date Current period created.
SIT-FA-DT	First use date.
SIT-FA-REL	First use JOB/SCAN release.
SIT-LA-DT	Last use date.
SIT-LA-REL	Last JOB/SCAN release used.
SIT-P	Number of periods the site has used.
TOT-EDIT	Total Edit macro JOB/SCAN runs.
TOT-ISPF	Total ISPF panel JOB/SCAN runs.
TOT-BAT	Total Batch JOB/SCAN runs.
TOT-JOBS	Total JOBS encountered.
TOT-PRCS	Total PROCs encountered.
TOT-STEP	Total Steps encountered.
TOT-DD	Total DDs encountered.
TOT-LINE	Total Lines scanned.
TOT-JCLE	Total JCL Errors.
TOT-STDE	Total Standards Errors.

*Period Fields*

Field	Description
	Status of the Period:
PER-STAT	<ul style="list-style-type: none"> <li>• INACTIVE for periods no longer actively receiving data.</li> <li>• CURRENT the active period.</li> <li>• DELETED the period is deleted.</li> </ul>
PER-CR-U	User ID that created period.
PER-CR-DT	Date Period created.
PER-CR-REL	JOB/SCAN release this period was. created under
PER-FA-DT	First use date.
PER-FA-REL	First use JOB/SCAN release.
PER-LA-DT	Last-used date or NO J/S RUN if JOB/SCAN has not been run since period created.
PER-LA-REL	Last JOB/SCAN release used.
PER-DE-U	User that deleted the period.
PER-DE-DT	Date the period was deleted.
TOT-EDIT	Total EDIT macro JOB/SCAN runs.
TOT-ISPF	Total ISPF panel JOB/SCAN runs.
TOT-BAT	Total Batch JOB/SCAN runs.
TOT-JOBS	Total JOBS encountered.
TOT-PRCS	Total PROCs encountered.
TOT-STEP	Total Steps encountered.
TOT-DD	Total DDs encountered.
TOT-LINE	Total Lines scanned.
TOT-JCLE	Total JCL Errors.
TOT-STDE	Total Standards Errors.



### User Fields

Field	Description
USR-FA-DT	First use date
USR-FA-REL	First use JOB/SCAN release
USR-LA-DT	Last use date
USR-LA-REL	Last JOB/SCAN release used
	User status:
USR-STAT	<ul style="list-style-type: none"> <li>• ACTIVE</li> <li>• INACTIVE</li> </ul>
TOT-EDIT	Total EDIT macro JOB/SCAN runs
TOT-ISPF	Total ISPF panel JOB/SCAN runs
TOT-BAT	Total Batch JOB/SCAN runs
TOT-JOBS	Total JOBS encountered
TOT-PRCS	Total PROCs encountered
TOT-STEP	Total Steps encountered
TOT-DD	Total DDs encountered
TOT-LINE	Total Lines scanned
TOT-JCLE	Total JCL Errors
TOT-STDE	Total Standards Errors

### Detail Fields

Field	Description
PERI	The period to which this record belongs.
DATE	The date of the run.
RLSE	The release of JOB/SCAN that made the record.

Field	Description
TYPE	The type of run (EDIT for Edit macro, ISPF for Panel, or BATCH for batch).
EXIT	The name of the standards exit.
JOBS	The number of jobs.
PROCS	The number of procs.
STEPS	The number of steps.
DD	The number of DDs.
LINES	The number of lines.
JCLE	The number of JCL errors.
STDE	The number of standards errors.

---

# User Information - JOB/SCAN

This documentation introduces the features and benefits of ASG-JOB/SCAN (herein called JOB/SCAN), explains how to initialize and execute the product, provides batch mode instructions, and describes JOB/SCAN's reports. It also provides detailed instructions on using Edit macros, JCL Change, Reformatter, the Standards Facility, and interfaces to other products.

Topic	Description
<a href="#">Introduction to JOB/SCAN</a>	Introduces the facilities of JOB/SCAN.
<a href="#">JCL Validation</a>	Describes the basic validation function.
<a href="#">Advanced Topics for JCL Validation</a>	Describes the advanced features of JOB/SCAN validation.
<a href="#">Reports</a>	Describes the reporting facilities.
<a href="#">JCL Change</a>	Describes the JCL changing facility.
<a href="#">JCL Reformatter</a>	Describes the JCL reformatting facility.
<a href="#">ROSCOE Interface</a>	Describes the use of JOB/SCAN in the ROSCOE RPF or ETSO environment.

---

# Introduction to JOB/SCAN

This topic introduces the facilities of JOB/SCAN.

- [JOB/SCAN Basics](#)
- [Initializing JOB/SCAN](#)

## JOB/SCAN Basics

This section gives a brief introduction to the JOB/SCAN modes and its facilities.

### JOB/SCAN Operating Modes

JOB/SCAN provides these modes for employing its capabilities:

- panel
- Edit macro
- batch

From each of these, you can access the total range of JOB/SCAN facilities.

### JOB/SCAN Facilities

JOB/SCAN from ASG helps data center personnel manage JCL with:

- Validation
- Reporting
- Change
- Reformatting
- Standards
- Roscoe

Each facility has its own control parameters.

### Validation

JOB/SCAN validates JCL in an input job stream so potential run-time errors can be corrected before the job stream is submitted for processing. JOB/SCAN processes the entire job stream, expands PROCs, resolves backward references, substitutes symbolics, applies overrides, and then evaluates each statement for correctness in the MVS environment.

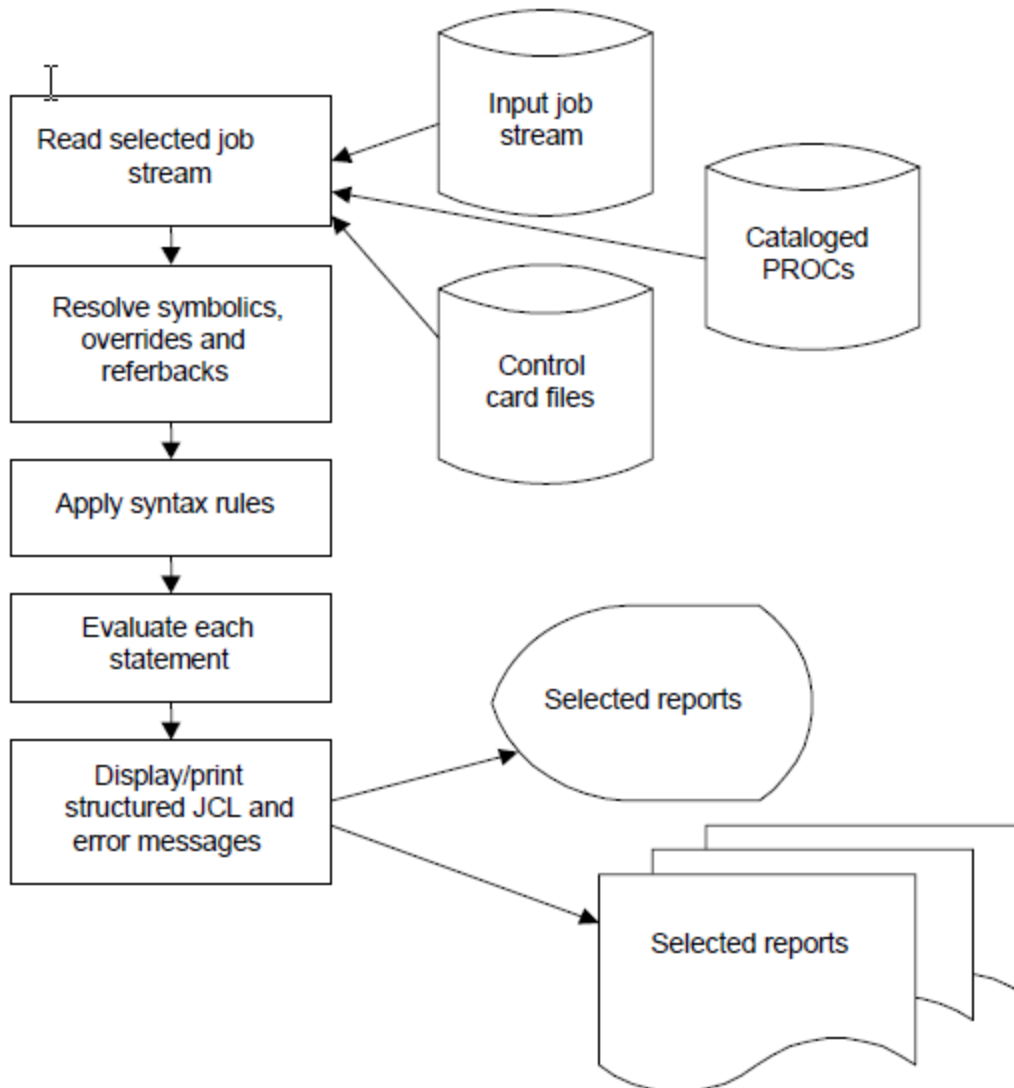
Optional functions of JOB/SCAN verify file access authority for the JOB/SCAN job, verify run-time access authority to datasets by dataset name, and enforce site standards. JOB/SCAN simulates the actual execution environment during the validation process.

These are major steps in the JOB/SCAN JCL Validation process:

- The JOB/SCAN user selects the JCL members to be validated.
- JOB/SCAN validates JCL members selected in Step 1, detecting all errors in one pass.
- JOB/SCAN displays any error messages in the Structured JCL Listing.

- The user corrects any errors.
- The entire job stream is ready for error-free execution.

#### The JOB/SCAN JCL Validation Process



#### Types of Errors Detected by JOB/SCAN

This table summarizes the types of errors that JOB/SCAN can detect.

Type of Error	Examples
Syntax	<ul style="list-style-type: none"> <li>• Format errors</li> <li>• Unbalanced parentheses.</li> <li>• Illegal keywords</li> <li>• Invalid parameters</li> </ul>

Type of Error	Examples	
Run-time	<ul style="list-style-type: none"> <li>Invalid backward references.</li> <li>Duplicate dataset names</li> <li>Dataset not found</li> <li>Program not found</li> <li>Invalid concatenation</li> </ul>	<ul style="list-style-type: none"> <li>Invalid JES2 or JES3 statements</li> <li>Incompatible DCB Parameters</li> <li>PDS member not found</li> <li>Missing UNIT or SPACE parameter</li> <li>LIKE dataset not found</li> </ul>
SMS	<ul style="list-style-type: none"> <li>Invalid DATACLAS, MGMTCLAS, or STORCLAS value</li> <li>Non-SMS dataset refers to SMS volume</li> </ul>	<ul style="list-style-type: none"> <li>JCL parameter inconsistent with ACS routine (SMS definition coded but not used)</li> </ul>
IDCAMS	<ul style="list-style-type: none"> <li>Invalid keywords</li> <li>Unbalanced parentheses</li> <li>Duplicate keyword parameters</li> </ul>	<ul style="list-style-type: none"> <li>Missing positional parameters</li> <li>Unresolved ddname references</li> </ul>
IMS	<ul style="list-style-type: none"> <li>Invalid PSB, DBD, or ACB. PSB, DBD, or ACB not found</li> </ul>	<ul style="list-style-type: none"> <li>Duplicate IMS parameter.</li> </ul>
DB2	<ul style="list-style-type: none"> <li>Invalid DB2 syntax keyword or parameter</li> <li>Required DB2 keyword not specified</li> </ul>	<ul style="list-style-type: none"> <li>DB2 library or program not found</li> </ul>
Site-specific standards as defined for the site.		
Miscellaneous	<ul style="list-style-type: none"> <li>Overrides out of order</li> <li>Duplicate DD statements</li> </ul>	<ul style="list-style-type: none"> <li>Duplicate symbolics</li> <li>Generation data group bases not found</li> </ul>

### Reporting Facilities

After each run, JOB/SCAN writes the *Structured JCL Listing* (SJL) report. This report is divided into several sections, with complete details on JCL errors found, statements changed, standards violations caught, and statements reformatted. See [Reports](#) for more about the SJL.

The *Usage Report*, contains machine-level and user-level usage, including number of uses of JOB/SCAN and the number of errors found, statements changed, standards violations caught, and statements reformatted. The [JOB/SCAN Usage Report Detail Statement](#) completely explains this report.

### JCL Change Facility

The JCL Change Facility (herein referred to as Change) makes the task of changing JCL easier.

Using a simplified English-like syntax, change commands to automate:

- Changing of certain items when jobs move from test to production.
- Correcting misspellings.
- Adding or changing of whole or partial parameters and subparameters.
- Modifying IBM standard parameters, user-defined parameters, and character strings.
- Modifying existing JCL (including adding and deleting entire JCL statements) according to new or changed site standards, coding rules, or changes in responsibility.

Change reads and validates the set of change commands, then applies them to specified JCL members. These commands can either be applied unconditionally or conditionally to the members, either in simulation or as output directed to a temporary dataset. From this dataset, members are validated and promoted. Change runs in batch and panel modes.



Change does not support modifications to variable-length, blocked record (RECFM=VB) datasets.

### Reformatter Facility

JOB/SCAN's Reformatter facility (herein referred to as Reformatter) restructures JCL into a standard, easy-to-read format based on built-in defaults or site specifications. After the format is specified, it can be used repeatedly to make all JCL easier to read, understand, and modify. Reformatter can be used to:

- Begin the operation field in a specific column.
- Specify the beginning column of the first parameter for each statement.
- Specify number of parameters per line and the order in which they appear.
- Re-sequence or eliminate sequence numbers in columns 73 through 80.
- Restructure or eliminate comment fields.
- Realign EXEC and DD parameters so that all JCL is formatted consistently.
- Restructure PROC and PEND JCL statements.

Reformatter does not require a command library member. In a JOB/SCAN run which exercises Reformatter execution only, and with no Reformatter member specified, the Reformatter uses B<sub>x</sub> table entries in the JOB/SCAN SJJSPARM library to reformat the JCL. If a member is specified, the Reformat commands override the B<sub>x</sub> table entries. (See [Accessing Reformatter through Tables.](#))



ASG recommends using JOB/SCAN to validate the JCL before reformatting to make sure that the JCL syntax is correct, because invalid JCL cannot be properly reformatted.

## Standards Facility

The Standards Facility (herein referred to as Standards) helps automate requirements for the content of JCL. It does not reformat JCL nor check for formatting errors or inconsistencies. Standards works with JOB/SCAN after it process each statement. JOB/SCAN calls user-written routines which supply information and SJL messages for statements that do not meet the standards. Also see [Standards Information](#) for more information about standards.

## Roscoe Facility

In the ROSCOE RPF environment, the ASG RPF facility provides RPF panels similar in organization to the ISPF panels. These panels select the reports to create and display, and the jobs to be scanned. Creation of reports and scanning of jobs are not available in foreground mode. However, this is accomplished by selecting batch execution via the panels. ROSCOE users can specify a ROSCOE library type in addition to other valid library types. This enables JOB/SCAN to accept JCL as input from a ROSCOE library.

See [ROSCOE Interface](#) for more information on the ROSCOE facility.

# Initializing JOB/SCAN

Because of the complexity of what it does, JOB/SCAN requires initialization of a few essential variables and screens. As users employ JOB/SCAN in their daily tasks, JOB/SCAN retains changes to initial settings in the profile.

## The First-time Use Initialization Sequence

The first time JOB/SCAN is accessed by a particular userid, the system steps through this sequence of four panels:

- The JOB/SCAN Welcome panel that explains initialization.
- The Hardcopy Output Characteristics panel.
- The Job Statement for Batch Submission panel.
- The Processing Options panel.

## The Welcome panel

This panel, briefly describes the initialization sequence.

- Press Enter to proceed through the initialization sequence.

Or

- Press End to cancel and return to the previous panel and initialize JOB/SCAN at a later time.



### The ASG-JOB/SCAN Welcome panel

```

* * * * *
*
* Welcome to ASG-JOB/SCAN JSX.X.X!
*
* ASG-JOB/SCAN has many functions which can execute in batch or write
* hardcopy output to SYSOUT while executing in the foreground.
* The following panels ask you to enter the information that
* ASG-JOB/SCAN needs to perform these and other functions.
*
* Each panel has help panels that explain how to use it.
* If you would like more information while filling in a panel,
* press the HELP key (usually PF1). If you are using ISPF 3.1 or
* later, field-level help is also available. For help on a field
* on a panel, place the cursor on the field and press the HELP key.
*
* After you have filled in the panels, the ASG-JOB/SCAN Primary Menu
* will be presented.
*
* Press ENTER to continue ASG-JOB/SCAN initialization.
* Press END to return to the previous menu.
*
*

```

### The Hardcopy Output Characteristics Panel

The second panel in the Initialization sequence is the **HARDCOPY OUTPUT CHARACTERISTICS** panel. It allows options to be designated for TSO hardcopy.

#### Hardcopy Output Characteristics Panel

```

----- HARDCOPY OUTPUT CHARACTERISTICS ----- ASG-JOB/SCAN
COMMAND  ==>

Force OPE hardcopy for Initialization ==> N (Y/N)

Enter/verify values for Hardcopy output:
Class      ==> A
Dest       ==>
Copies     ==> 1      1-255
FCB        ==>
Forms      ==>
UCS        ==>      (For other than IBM 3800)
Chars      ==>      (Only for IBM 3800)
FSSData    ==>
FormDef    ==>
PageDef    ==>

Hold       ==> Y      (Y/N)

If you wish to be prompted with this panel when the Hardcopy option
is specified for foreground options, enter Y below:
Hardcopy panel prompt ==> N (Y/N)

```

### Using This Panel

See [Panel Fields](#) for information on fields on this panel.

#### To verify entries in this panel

1. Press Enter.

If any invalid entries are found, an error message appears in the upper right of the panel and the cursor moves to the field in error.

2. Correct the error and press Enter again.
3. Repeat steps 1 and 2 until no more errors are found.
4. Press End to proceed to the Job Statement for Batch Submission panel.

### Panel Fields

This table explains the required and optional fields of the Hardcopy Output Characteristics panel. In most cases, each value corresponds to an equivalent JCL parameter.

Field	Required/ Optional	Description
FORCE OPE HARDCOPY FOR INITIALIZATION	Optional	Tells whether or not to print a hardcopy of the Options in Effect report each time JOB/SCAN is executed. The report shows the Run and Report Options in effect. Type Y for yes or N for no; the default is N.
CLASS	Required	Designates the output class for hardcopy output. Because JOB/SCAN cannot verify the class before execution, make sure the class is valid for the site.
DEST	Optional	Specifies a destination value for hardcopy output. This field is compatible with JES length and structure.
COPIES	Optional	Specifies the number of copies (from 1 to 255 for JES2, or 1 to 254 for JES3) for hardcopy output. The default is 1.
FCB	Optional	Specifies the name of the forms control buffer image JES is to use to guide printing of this SYSOUT dataset.
FORMS	Optional	Designates special forms to be used for hardcopy output requested during foreground processing.
UCS	Optional	Used for printers other than the IBM 3800. This parameter is mutually exclusive with the Chars field.
CHARS	Optional	Used only if an IBM 3800 graphics font is installed. This parameter is mutually exclusive with the UCS field.
FSSDATA	Optional	The FSSDATA parameter is a subsystem-defined parameter to pass from a spooling product to a functional subsystem application (FSA) or other despooler. It is not necessary to

Field	Required/Optional	Description
		enclose this parameter in apostrophes. If you normally enter an FSSDATA parameter on an OUTPUT statement in your JCL, enter the FSSDATA value here. Otherwise, leave this field blank.
FORMDEF	Optional	Identifies a library member that contains statements to tell the Print Services Facility (PSF) how to print the SYSOUT dataset on a page-mode printer.
PAGEDEF	Optional	Identifies a library member that contains statements to tell the Print Services Facility (PSF) how to print the SYSOUT dataset on a page-mode printer.
HOLD	Optional	Tells whether or not to hold SYSOUTs for TSO/E output processing. Type Y for yes, or N for no.
HARDCOPY PANEL PROMPT	Optional	Type Y if need panel to appear when specify Hardcopy output on a Foreground execution panel. Otherwise, leave default of N.

### The Job Statement for Batch Submission Panel

The third panel in the initialization sequence is the Job Statement for Batch Submission panel. It specifies the information for panel-driven batch requests:

- A Job statement for batch execution.
- Step accounting information.
- Whether to be prompted with this panel every time batch is run.

### Job Statement for Batch Submission Panel

```

----- JOB STATEMENT FOR BATCH SUBMISSION ----- (2 of 2)
COMMAND  ===>

JOB STATEMENT INFORMATION: (Required - Enter/Verify JOB control statements)
===> //ASGUSR1 JOB (ACCOUNT),'USER NAME          ',
===> //      NOTIFY=ASGUSR
===> //*
===> //*
===> //*

Step accounting information for generated EXEC statement (61 chars max)
ACCT  ===>

If you wish to be prompted with this panel when the Batch option
is specified, enter Y below:
JOB Statement panel prompt  ===> N (Y/N)

Press <END> to return.

```

## Using This Panel

### To fill out this panel

- Follow the guidelines in [Completing Job Statements Section: Guidelines](#) and [Completing Step Accounting Section: Guidelines](#)

### To verify entries in this panel

1. Press Enter.

If any invalid entries are found, an error message appears in the upper right of the panel, and the cursor moves to the field in error.

2. Correct the error and press Enter again.
3. Repeat steps 1 and 2 until no more errors are found.
4. Press End to return to the ASG-JOB/SCAN Primary Menu panel.

## Completing Job Statements Section: Guidelines

When filling in the job statement lines, follow these guidelines:

- All five lines of this section must be coded, even if some are comment lines.
- Lines may consist of a Job statement, JES statements, or any other statements that are valid before the first EXEC statement in a job. These lines are inserted before an execute PROC statement, which is generated when a batch run is selected from the panels.
- The information must conform to valid JCL, JES2, or JES3 syntax rules.
- If a job name is not coded, it defaults to the TSO userid with the suffix 1 the first time JOB/SCAN is executed.
- Use site conventions for specifying the correct job name, accounting field, and other JCL parameters.

## Completing Step Accounting Section: Guidelines

When filling in the ACCT field, follow these guidelines:

- Specify up to 61 characters for the ACCT= parameter on the EXEC statement to be generated for batch execution.
- Leave this field blank if step accounting is not being used.

## Forcing the JOB Statement Prompt

By typing Y at the JOB Statement panel prompt field, JOB/SCAN displays this panel every time JOB/SCAN is executed in panel-driven batch mode. If N is specified, JOB/SCAN does not display this panel. After initialization, you can access the panel from the ASG-JOB/SCAN Primary Menu panel.

## The Processing Options Panel

The fourth panel in the initialization sequence is the Processing Options panel. When JOB/SCAN is executed using ISPF panels, the values entered on the Processing Options panel override parameters coded in Run Options in members in the JOB/SCAN SJJSPARM library. This allows for tailoring of Run Options by each user as necessary. See [Panel Fields](#) for information on fields of this panel.

**Processing Options Panel**

```

----- JOB/SCAN PROCESSING OPTIONS -----
COMMAND ==>
Use Error Table          ==> N          (Y/N)
Condition Code Processing ==> N          (Y/N)
Catalog Access          ==> Y          (Y/N)
VTOC Access              ==> Y          (Y/N)
PDS Member Verification ==> Y          (Y/N)
Security Exit DSNAMES Check ==> Y      (Y/N)
Program Checking         ==> Y          (Y/N)
Disable SMS Processing   ==> N          (Y/N)
IDCAMS Processing       ==> Y          (Y/N)
Use SCHEDULER interface ==> N          (Y/N)
Job Limit                ==>           (1 to 999 or Blank For No Limit)
Access Control Cards     ==> Y          (Y/N)
Control Card Limit       ==>           (1 to 999 or Blank For No Limit)
Subsystem                ==> JES2      (JES2/JES3/None)
Suppress Error Messages  ==> NO       (A,W,E,0 to 99,or NO)
Extended Error Text      ==> S          (X=Extended,S=Short)
    
```

**Using This Panel**

**To validate the entries of this panel**

1. Press Enter to verify entries.

If any invalid entries are found, an error message appears in the upper right of the panel, and the cursor moves to the field in error.

2. Correct the error and press Enter again.
3. Repeat steps 1 and 2 until no more errors are found.
4. Press End to proceed to the ASG-JOB/SCAN Primary Menu panel.

**Panel Fields**

This table explains the fields of the Processing Options panel. Each field corresponds to a Run or Report option.

Field	Default	Description
Use Error Table	N	Type Y to access Control Table AW for error suppression based on severity levels.
Condition Code Processing	N	Type Y to request that JOB/SCAN evaluate all COND= parameters and // IF statements, and process the JCL accordingly. Entering Y also uses Control Tables AU and AR for step processing and return code simulation.
Catalog Access	Y	Type N to inhibit access to the system catalog.
VTOC Access	Y	Type N to inhibit access to the DASD VTOCs.

Field	Default	Description
PDS Member Verification	Y	Type Y to verify the existence of PDS members referenced in the JCL. If Y is entered, the VTOC Access parameter must also be set to Y. Verification is not done for: - datasets with a disposition of NEW. - Control card datasets if the Access Control Cards parameter is Y, and a JLJPF DD statement has been coded in the JOB/SCAN JCL, TSO logon PROC, or on a File Allocations Run Option statement.
Security Exit Dsname Check	N	Type Y if the security exit is to check each dataset name in the input JCL for possible security violations. If there is no security exit active or no DSNNAME-checking support in the exit, this option is ignored.
Program Checking	Y	Type Y to cause JOB/SCAN to validate the existence of programs. N turns off this validation. Coding Y for this parameter and N on either Catalog Access or VTOC Access, will result in a volume mount request if the JOBLIB or STEPLIB volume is not on-line to the system.
Disable SMS Processing	N	Type Y to bypass all SMS processing. If N is specified, options specified in the SMS Run Option are to be in effect for this JOB/SCAN run. This variable is ignored if the SMS subsystem is not active during JOB/SCAN.
IDCAMS Processing	Y	Type Y to cause validation to be performed on IDCAMS control statements. N turns off this validation. The Catalog Access and VTOC Access parameters must be set to Y if IDCAMS statements are not in-stream.
Use SCHEDULER Interface	N	Type Y to turn on the scheduler interface specified by the SCHEDULER run option. Type N to disable scheduler interfaces. The use of this option is explained in <a href="#">User Information - JOB/SCAN with Schedulers</a> .
Job Limit	0 [no limit]	Type the maximum number of jobs to be processed (1-999) in a single JOB/SCAN execution. (See <a href="#">Job Limit</a> .)
Access Control Cards	Y	Type Y to access control card libraries, verify control card input, and include control cards in the Structured JCL Listing (SJL). The Catalog Access and VTOC Access parameters must

Field	Default	Description
		be set to Y if this parameter is set to Y. If set to Y, dataset names in Control Table AQ or ddnames in Control Table AB are treated as control card input. Any dataset names in Control Table CG will not be expanded in the SJL.
Control Card Limit	0 [no limit]	Limits the number of control cards (1-999) listed in the SJL for each ddname or dataset name identified as control card input. This parameter is ignored if Access Control Cards is set to N.
Subsystem	No default	Use JES2, JES3, or NONE to indicate whether the JCL contains JES2 or JES3 statements to be validated for syntax. An entry of NONE tells JOB/SCAN not to evaluate JES statements.
Suppress Error Messages	NO	Use NO to include all messages in the SJL. Type A to suppress advisory messages; W to suppress warning and advisory messages; or E to suppress advisory, warning, and error messages. Alternatively, use a number from 0 to 99 to suppress all error messages with a severity level equal to or less than the number coded. This option is independent of Control Table AW, which can suppress specific errors.
Extended Error Text	S	Type X to cause the extended error message text to be printed for each error listed on the SJL. S causes only the short error message to be included.

**Corresponding Option Parameters**

When running JOB/SCAN from the Validation panel, Report Option and Run Option values from fields of the JOB/SCAN Processing Options panel override the corresponding parameters from the SJJSPARM library options members. [Relating JCL Validation Panel Fields to SJJSPARM Options](#) shows the relationships between the panel fields to the statement parameters.

**Changing Field Values in the Initialization Panels**

JOB/SCAN saves the information users enter into the fields of the initialization (and other product panels) from session to session in the ISPF profile. For example, when printing reports, JOB/SCAN uses the output class for hardcopy output from the HARDCOPY OUTPUT CHARACTERISTICS panel. This table explains how to access these panels after the initialization period:

Panel	How to Change
Hardcopy Output	Select option 0 (Hardcopy Output and JOB Statement Definition) from the ASG-

Panel	How to Change
Characteristics	JOB/SCAN Primary Menu panel.
Job Statement for Batch Submission	Select option 0 (Hardcopy Output and JOB Statement Definition) from the ASG-JOB/SCAN Primary Menu panel, and press End.
Processing Options	Select Processing Options from the ASG-JOB/SCAN Primary Menu panel, or type P on the command line of the JCL Validation panel.

## Primary Menu

After you complete the initialization panels, JOB/SCAN displays the ASG-JOB/SCAN Primary Menu. This panel provides various options for validating JCL, changing and/or reformatting JCL, specifying processing options, and creating/modifying selection lists.

### JOB/SCAN Primary Menu

```

----- ASG-JOB/SCAN PRIMARY MENU -----
SELECTION ==>

  0 - Hardcopy Output and JOB Statement Definition   USERID - CS081A
  1 - Validate JCL                                  PREFIX  - CS081A
  2 - Change / Reformat JCL                         DATE    - 09/20/2016
  3 - Processing Options                            TIME    - 09:58
  4 - Selection List Maintenance                    APPLID  - JS
  X - Exit

```



# JCL Validation

This topic describes the basic validation function.

- [The Basic Validation Process](#)
- [Validating JCL from the Panels](#)
- [Validating JCL with Edit Macros](#)
- [Validating JCL in Batch](#)

## The Basic Validation Process

The major steps in the JOB/SCAN JCL validation process for each mode are:

1. Select the JCL members to be validated (in Edit macro mode, this is the member in Edit/View; in batch this is accomplished with the SELECT= statement).
2. Select the mode in which to run validation: panels, Edit macros, batch.
3. Start JOB/SCAN to:
  - validate JCL members selected, detecting all errors in one pass.
  - write any error messages to the Structured JCL Listing.
4. View the SJL and notes any errors for correction.
5. Correct the original JCL. (The Edit macros present the errors in the context of the original JCL so that error-correction is expedited.)
6. Revalidate any members you change. When the SJL reports that JOB/SCAN found no errors, the selected members are error-free and ready for the next step in data center operations.

## Validating JCL from the Panels

You execute JCL validation in panel-mode from the JOB/SCAN JCL Validation panel.

### *To access the JCL Validation Panel*

- Select Validate JCL from the JOB/SCAN Primary Menu.

JOB/SCAN displays JCL Validation panel.

**JOB/SCAN - JCL Validation Panel**

```

----- JOB/SCAN - JCL VALIDATION -----
COMMAND ==>

      F - Foreground execution mode          P - Processing Options
      B - Batch execution mode              E - Edit JCL member
      S - Submit After Validation           BR - Browse JCL member
Input Library  ==> 'ASGCOM2.BOOK.JOBS'
Input Type    ==> J          (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ==> PDS       (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==>
Exclude Members ==>
First Proclib ==>
Proclib Type  ==>          (PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
Report Heading ==>
Display        ==> Y (Yes/No/ErrorsOnly)  Browse/View ==> B (B/V)
Hardcopy       ==> Y (Yes/No/ErrorsOnly/SpaceOnly)
Options in Effect Listing ==> N (Y/N)      SJL PDSOUT ==> N (Y/N)

Selection Exit ==>          Parm ==>
Standards Name ==>          Parm ==>

```

Use this panel to specify these validation options:

- The members to be validated, their library and its type.
- The first PROC library JOB/SCAN should examine for PROCs.
- Whether standards checking is done.
- Whether the task is done in foreground (your TSO session is occupied waiting for results) or in background (your TSO session is free to start other tasks).
- SJL components and an optional heading.

**Selecting Members**

The JOB/SCAN - JCL Validation panel fields, Select Members and Exclude Members, provide a flexible means of selecting members of the dataset specified in the Input Library field for processing by JOB/SCAN.

**Selecting jobs on the JCL Validation Panel**

```

...
Input Library  ==> BOOK.JOBS
Input Type     ==> J          (J - JOB JCL, P - PROC, or L - Selection List)
Library Type   ==> PDS       (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> JOB001,MYJOB02-MYJOB10
Exclude Members ==> MYJOB05
...

```

You may select members in one or a combination of these ways:

- One or more *specific* member *Names*.

```
Select Members ==> JOB003, JOB729, JOB018, JOB729
```

Members specifically selected by name cannot be excluded by entries in the Exclude Members field.

- Member names included within a specified sort *Range*.

```
Select Members ==> JOB000-JOB008
```

```
Exclude Members ===> JOB004-JOB006
```

- Member names beginning with the same characters (*Prefix*).

```
Select Members ===> PREFIX=JOB0
```

This is analogous to *JOB0\**.

- Member names that fit a *Mask* of matching and wildcard characters (\*).

```
Select Members ===> **B0*3**
```

- *ALL* of the member names.

```
Select Members ===> ALL
```

- Any combination of the above:

```
Select Members ===> MYJOB,*YZ****,ALL
```

[Selecting Members By Name from the Input Library Dataset](#), describes these options in detail in the context of submitting a JOB/SCAN JCL Validation run from the JOB/SCAN - JCL Validation panel.



If you want JOB/SCAN to automatically prepend your user ID as a high-level-qualifier, type unquoted values into the Input Library and First Proclib fields. Otherwise, place quotes around the library name to fully specify a library and JOB/SCAN uses them as-is.

## Selecting the Library and Its Type

For setting the input library and its type, put the name of the library in the Input Library field enclosed in quotes.

### Setting up the input library JCL Validation panel

```
...
Input Library   ===> 'ASGCOM2.BOOK.JOBS'
Input Type      ===> J           (J - JOB JCL, P - PROC, or L - Selection List)
Library Type    ===> PDS         (PDS, PAN, LIB, LIB-3, or SEQ)
...
```

See [Selecting Members By Name from the Input Library Dataset](#) for more about selecting a library.

## Setting up the First PROCLIB

For setting up a first PROC library, type the dataset name and type for a PROC library that you want JOB/SCAN to search first.

### Setting up the PROC library searching from the JCL Validation panel

```
...
First Proclib   ===> OURLIB
Proclib Type    ===> PDS         (PDS, PAN, LIB, or LIB-3)
...
```

See [Specifying a First Proclib for JCL Validation](#) for complete details.

## Setting up the SJL

### To set up the SJL

- Type an optional heading and values for displaying or printing the report in the appropriate fields. Also, you may include the Options in Effect Report (OPE).

### Setting up the SJL from the JCL Validation panel

```

...
Structured JCL Listing:
  Report Heading  ===> Daily JCL Validation Report for XYZ Company
  Display         ===> Y (Yes/No/ErrorsOnly)  Browse/View ===> B (B/V)
  Hardcopy       ===> Y (Yes/No/ErrorsOnly/SpaceOnly)
  Options in Effect Listing  ===> N (Y/N)          SJL PDSOUT ===> N (Y/N)
...

```

See [Structured JCL Listing \(SJL\)](#) for complete details. See [Viewing the SJL](#) for brief instructions on viewing the SJL.

## Setting up Standards

### To set up JOB/SCAN to enforce standards

- Type the name of the standards program in the Standards name field.

### Setting up standards from the JCL Validation panel

```

...
Selection Exit  ===> SEL01      Parm  ===>
Standards Name  ===> STD01      Parm  ===>

```

See [Standards Information](#) for complete details.

## Selecting the Action

### Foreground Execution

#### To validate JCL in foreground mode

- Type F on the command line and press Enter.

### Executing JOB/SCAN in foreground

```

----- ASG-JOB/SCAN - JCL VALIDATION -----
COMMAND ===> F

      F - Foreground execution mode          P - Processing Options
      B - Batch execution mode              E - Edit JCL member
      S - Submit After Validation           BR - Browse JCL member
Input Library  ===> CS082B.JCL
Input Type    ===> J                      (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ===> PDS                    (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ===> BR1*
Exclude Members ===>
First Proclib ===>
Proclib Type  ===>                      (PDS, PAN, LIB, or LIB-3)
...

```

After JOB/SCAN completes, it displays the SJL according to the SJL reporting options set on this panel.

### Background Execution (Called Batch on this panel)

#### To validate JCL in background mode

- type B on the command line and press Enter.

#### Executing JOB/SCAN in background

```

----- ASG-JOB/SCAN - JCL VALIDATION -----
COMMAND ==> B

      F - Foreground execution mode          P - Processing Options
      B - Batch execution mode              E - Edit JCL member
      S - Submit After Validation           BR - Browse JCL member
Input Library ==> CS082B.JCL
Input Type   ==> J          (J - JOB JCL, P - PROC, or L - Selection List)
Library Type ==> PDS       (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> BR1*
Exclude Members ==>
First Proclib ==>
Proclib Type ==>          (PDS, PAN, LIB, or LIB-3)
...

```

JOB/SCAN displays a system message at the bottom of the Validation panel.

#### System message for background

```

...
Selection Exit ==>          Parm ==>
Standards Name ==>          Parm ==>
IKJ56250I JOB ASGUSR1 (JOB09588) SUBMITTED
***

```

Use any convenient method to view SDSF output, particularly the JLKAC.

### Submit After Validation

#### To submit the job directly after validation

1. Type s on the command line and press Enter.

#### Submitting a job from the JCL Validation panel

```

----- ASG-JOB/SCAN - JCL VALIDATION -----
COMMAND ==> S

      F - Foreground execution mode          P - Processing Options
      B - Batch execution mode              E - Edit JCL member
      S - Submit After Validation           BR - Browse JCL member
Input Library ==> CS082B.JCL
Input Type   ==> J          (J - JOB JCL, P - PROC, or L - Selection List)
Library Type ==> PDS       (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> BR1*
Exclude Members ==>
First Proclib ==>
Proclib Type ==>          (PDS, PAN, LIB, or LIB-3)
...

```

JOB/SCAN validates the member. If it finds errors it displays the SJL. You can then decide whether the errors need to be addressed.

2. Press End. If JOB/SCAN did not find errors, it displays the Conditional Job Submission panel.
3. Follow the instructions on the panel to submit or cancel the job.

#### JOB/SCAN - Conditional Job Submission panel

```

----- CONDITIONAL JOB SUBMISSION ----- ASG-JOB/SCAN
COMMAND ==> SUB

      Do you wish to submit this job stream
      for execution?

      - To SUBMIT the scanned JCL even though it had errors
      type SUB in the COMMAND line and press <ENTER>.

      - To RETURN to the ASG-JOB/SCAN panel without submitting,
      type CAN in the COMMAND line and press <ENTER>.

      or

      - Press END.

```

JOB/SCAN returns this message:

#### The system message from the Submit command

```

Selection Exit ==>          Parm ==>
Standards Name ==>        Parm ==>
IKJ56250I JOB ASGUSR1 (JOB09588) SUBMITTED
***

```

Use the usual method of viewing SDSF output used at your site, particularly the JLKAC.

### Setting/Viewing Processing options

#### *To view or set certain, listed processing options*

1. Type P on the command line and press Enter.

### Setting procession options from the Processing Options panel

```

----- ASG-JOB/SCAN PROCESSING OPTIONS -----
COMMAND ==>

  Use Error Table           ==> N           (Y/N)
  Condition Code Processing ==> N           (Y/N)
  Catalog Access           ==> Y           (Y/N)
  VTOC Access              ==> Y           (Y/N)
  PDS Member Verification  ==> Y           (Y/N)
  Security Exit DSNAMES Check ==> N       (Y/N)
  Program Checking         ==> Y           (Y/N)
  Disable SMS Processing   ==> N           (Y/N)
  IDCAMS Processing        ==> Y           (Y/N)
  Use SCHEDULER interface  ==> Y           (Y/N)
  Job Limit                ==>           (1 to 999 or Blank For No Limit)
  Access Control Cards     ==> Y           (Y/N)
  Control Card Limit       ==>           (1 to 999 or Blank For No Limit)
  Subsystem                ==> JES2       (JES2/JES3/None)
  Suppress Error Messages  ==> NO        (A,W,E,0 to 99,or NO)
  Extended Error Text      ==> S         (X=Extended,S=Short)

Press <ENTER> to verify option values or <END> to leave and save option values.

```

### Edit/Browse a Member

#### To edit or browse JCL

- Type E or BR respectively on the command line and press Enter.

### Editing or Browsing JCL from the JCL Validation panel

```

----- ASG-JOB/SCAN - JCL VALIDATION -----
COMMAND ==> BR

  F - Foreground execution mode      P - Processing Options
  B - Batch execution mode           E - Edit JCL member
  S - Submit After Validation        BR - Browse JCL member

Input Library ==> CS082B.JCL
Input Type    ==> J                 (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ==> PDS               (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> BR1*
Exclude Members ==>
First Proclib ==>
Proclib Type  ==>                 (PDS, PAN, LIB, or LIB-3)
...

```

### Viewing the SJL

In panel mode, JOB/SCAN displays the SJL after it completes the validation. To view the SJL at a later time, go to the held output, and view the JLKAC dd.

## Validating JCL with Edit Macros

Edit macros are commands that can be executed during an ISPF Edit session. JOB/SCAN provides several Edit macros to facilitate developing JCL. The Edit macros give comprehensive feedback about any errors found. This provides the opportunity to correct errors before leaving the ISPF Edit session.

This table summarizes the JCL Validation Edit macros provided with JOB/SCAN.

Macro	Description
JEM	Validates the JCL being edited. When complete, returns to edit mode with an error summary displayed at the top of the screen. <a href="#">JEM Edit Macro</a> provides a detailed description of JEM.
JMSG	Used following JEM execution to display extended error message text. <a href="#">JMSG: Extended Error Messages for the JEM Edit Macro</a> provides a detailed description of JMSG.
JSCAN	Validates the JCL being edited. Displays a Structured JCL Listing in browse mode. <a href="#">JSCAN Edit Macro</a> provides a detailed description of JSCAN.
JSESP	Runs JOB/SCAN resolving ESP variables. The use of this Edit macro is fully described in <a href="#">User Information - JOB/SCAN with ESP</a> documentation.
JSUB	Validates the JCL being edited. Displays a Structured JCL Listing in browse mode. If no errors are found, submits the JCL being edited to MVS for execution. <a href="#">JSUB Edit Macro</a> provides a detailed description of JSUB.
JTWS	Runs JOB/SCAN resolving TWS variables. The use of this Edit macro is fully described in <a href="#">User Information - JOB/SCAN with TWS</a> documentation.



If you code MEMBER=JSOPTXYZ in a JOBSCAN detail statement in the ACCESS CONTROL section of JSOPT01, this value overrides the OPT03 (membername) parameter on the JEM and JSCAN Edit macros.

### Optional Edit Macro Parameters

JOB/SCAN provides optional keyword parameters for the Edit macros JEM and JSCAN. Enter the parameters on the command line after *JEM* or *JSCAN*, in any order, separated by blanks.



When using OPT03 or OPT04, specify your user ID in the Options statement of the Access Control Statement in JSOPT01.

This table summarizes the Edit macro parameters for JEM and JSCAN and lists the abbreviated forms of the keywords, where applicable:

Parameter	Abr.	Description
SUB	n/a	Submits the JCL to MVS for execution if no validation errors are detected.



Parameter	Abr.	Description
COND ( <i>jobname. jobstep.stepname. returncode</i> )	C	Simulates a return code for the step specified. Use asterisks in place of values that do not matter. For more information, see the descriptions of <a href="#">Control Table AR: Return Code Simulation</a> and <a href="#">Processing Specifications (PSS)</a> run option.
STDPGM ( <i>pgmname</i> )	ST	Provides the name of a standards program to enforce site-specific standards. This parameter can be over-ridden by an Access Control Run Option statement. For more information, see <a href="#">Run Option Statements</a> .
PARM ( <i>parm</i> )	P	Specifies a parameter to be passed to the standards program.
OPT03 ( <i>membername</i> )	n/a	Specifies an alternate SJJSPARM library member to be used instead of JSOPT03. This parameter can be over-ridden by an Access Control Run Option statement. For more information, see <a href="#">Reference Information</a> .
OPTM ( <i>membername</i> )	n/a	Specifies an alternate SJJSPARM library member to be used instead of JSOPTM. This parameter can be overridden by an Access Control Run Option statement. For more information, see <a href="#">Reference Information</a> .
SELEXIT ( <i>pgmname</i> )	SE	Provides the name of a selection exit program to preprocess the JCL before JOB/SCAN validates it. For more information, see <a href="#">Selection Exit Interface</a> .
SELPARM ( <i>parm</i> )	SP	Specifies a parameter to be passed to the selection exit program.
DISPLAY (B) or DISPLAY (V)	D	(For JSCAN only) Displays the JSCAN Structured JCL Listing (SJL) in Browse mode (B) or View mode (V). The display mode is stored in an ISPF profile variable and is used on subsequent JSCAN invocations. The default display is Browse.

### JEM Edit Macro

The JEM Edit macro invokes JOB/SCAN JCL Validation on the JCL being edited. When complete, it returns to Edit mode, and the JCL statements can be revised in response to reported errors displayed at the top of the screen.

#### To use the JEM Edit macro to validate and correct JCL

1. While editing the JCL in ISPF Edit, type JEM on the command line and press Enter.

JOB/SCAN validates and re-displays the JCL in Edit mode, with error messages included as ISPF MSG lines. (See [JMSG: Extended Error Messages for the JEM Edit Macro](#) for information about using the JMSG Edit macro to display extended error message text for JEM error messages.)

2. Edit the JCL as needed to correct any detected errors.
3. Repeat Steps 1 and 2 until the JCL is free of errors.
4. Press End to terminate the ISPF Edit session.

## JEM JCL Display

In the JEM JCL display, the Error Summary appears at the top of the JCL. In-stream PROC errors are shown within the PROC. Cataloged PROCs and control cards are included as NOTE lines. Errors and the Error Summary are included as MSG lines.

A JEM Label appears on any JCL line containing an error and is associated with the error messages pertaining to that line. JEM Labels are in the format `.Jxxx`, where `xxx` is any alphanumeric characters.



- To make changes to cataloged PROCs and control card members, edit them separately from the JCL member being edited.
- Lines in the JEM JCL display designated NOTE or MSG cannot be edited.
- Edit macros do not generate held output.
- To remove NOTE or MSG lines from the display, type RESET on the command line and press Enter.

You may change any original line in this display to correct the error described by the MSG or `.Jaaa` notation. Then, you may revalidate this JCL with JEM directly from this panel.

**JEM JCL Listing**

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          ASGCOM2.TEST.CNTL (TESTER) - 01.09          Columns 00001 00072
Command ==>                                     Scroll ==> CSR
***** ***** Top of Data *****
==MSG> *** ERROR SUMMARY AND COUNTS *** JEM 6.1.2F 07/14/95 09:25:51
==MSG> 1 ADVISORY LEVEL
==MSG> 0 WARNING LEVEL
==MSG> 1 ERROR LEVEL
==MSG> 2 TOTAL ISSUED 0 SUPPRESSED.
==MSG> THE HIGHEST SEVERITY CODE ENCOUNTERED WAS 08.
==MSG> LABEL SV MSG.NO. ERROR MESSAGE
-----
==MSG> .JAAA 8 DSS4757E - READ ACCESS TO DATA SET DENIED TO USER ASGCOM2
==MSG> .JAAA 0 DSS8900A - DSN = "DSSI.$JS612F.ASM "
000100 //TESTER JOB (ACCTG), 'PRGRMR NAME', CLASS=A
000200 //PROCLIB DD DSN=ASGCOM2.TEST.PROCLIB, DISP=SHR
.JAAA //STEP01 EXEC PROC067
=NOTE= P1//PROC067 PROC
=NOTE= P1//PSTEP1 EXEC PGM=IEFBR14 PROC067 - VERSION 0000
=NOTE= P1//DD1 DD DISP=SHR, DSN=DSSI.$JS612F.ASM(SAF2)
==MSG> ***ERROR - DSS4757E - READ ACCESS TO DATA SET DENIED TO USER ASGCOM2
==MSG> *ADVISORY - DSS8900A - DSN = "DSSI.$JS612F.ASM "
=NOTE= P1//DD2 DD DUMMY
=NOTE= P1//DD3 DD DUMMY

```

**JSCAN Edit Macro**

The JSCAN Edit macro invokes JOB/SCAN JCL Validation on the JCL being edited. When complete, it displays a Structured JCL Listing in ISPF Browse mode.

See [Optional Edit Macro Parameters](#) for a list of the parameters available for use with this Edit macro.

**To use the JSCAN Edit macro**

1. While editing the JCL in ISPF Edit, type JSCAN on the command line and press Enter. JOB/SCAN validates the JCL and displays the Structured JCL Listing in ISPF Browse mode.
  2. Note the errors displayed in the Structured JCL Listing so they can be corrected in ISPF Edit mode.
- For complete information about reading the Structured JCL Listing (SJL), see [Reports](#).
3. Press End to return to ISPF Edit mode on the JCL member.
  4. Edit the JCL as needed to correct any detected errors.
  5. Repeat Steps 1 through 4 until the JCL is free of errors.
  6. Press End to terminate the ISPF Edit session.

JOB/SCAN displays the SJL.

**JSCAN Structured JCL Listing**

```

Menu  Utilities  Compilers  Help
-----
BROWSE      SYS95262.T134933.RA000.ASGUSR.JLSCRN.H01  Line 00000000 Col 001 078
***** Top of Data *****
JOB/SCAN    STRUCTURED JCL LISTING      6.1.2F      09/19/95  13:49:32
-----
//TESTER   JOB (ACCTG), 'PRGRMR NAME', CLASS=A                000100
//PROCLIB  DD   DSN=ASGUSR.TEST.PROCLIB, DISP=SHR              000200
//STEP01   EXEC PROC067                                       000300
P1//PROC067  PROC                                           00
P1//PSTEP1  EXEC PGM=IEFBR14          PROC067 - VERSION 0000  00
P1//DD1     DD   DISP=SHR, DSN=DSSI.$JS612F.ASM (SAF2)        00
***ERROR - DSS4050E - DATA SET NOT FOUND IN CATALOG
*ADVISORY - DSS8900A - DSN = "DSSI.$JS612F.ASM (SAF2) "
P1//DD2     DD   DUMMY                                         00
P1//DD3     DD   DUMMY                                         00
*** ERROR SUMMARY AND COUNTS ***
  1  ADVISORY LEVEL
  0  WARNING LEVEL
  1  ERROR LEVEL
  2  TOTAL ISSUED          0 SUPPRESSED.
      THE HIGHEST SEVERITY CODE ENCOUNTERED WAS      08.
REL LINE SV MSG.NO.      ERROR MESSAGE
-----
  8      8 DSS4050E - DATA SET NOT FOUND IN CATALOG
  9      0 DSS8900A - DSN = "DSSI.$JS612F.ASM (SAF2) "
...

```

**JSUB Edit Macro**

The JSUB Edit macro validates the JCL being edited and submits it to MVS for execution if it finds no errors. If it finds errors, JSUB displays the Structured JCL Listing in ISPF Browse mode, so that errors can be noted for correction. After displaying the Structured JCL Listing, JSUB displays a Conditional Job Submission panel, giving the choice of submitting the job (with errors uncorrected) or returning to ISPF Edit to correct the errors.

**To use the JSUB Edit macro**

1. While editing the JCL in ISPF Edit, type JSUB on the command line and press Enter. JOB/SCAN validates the JCL being edited, with either of these results:

- If it finds no errors, JSUB submits the JCL to MVS for execution, and remains in ISPF Edit mode on the JCL. Skip to Step 5.

**Or**

- If it finds errors, JSUB displays a Structured JCL Listing (similar to the JSCAN Structured JCL Listing shown in figure - [JEM JCL Listing](#)) in ISPF Browse mode.

2. Note the errors for correction when JSUB returns to ISPF Edit on the JCL.

See [Introduction to JOB/SCAN](#) for complete information about reading the Structured JCL Listing.

3. Press End to proceed to the JSUB Conditional Job Submission panel.
4. On the JSUB Conditional Job Submission panel, choose one of these actions:

- To submit the scanned JCL even though it has errors, type SUB on the command line and press Enter.

**Or**

- To return to ISPF Edit without submitting the JCL to MVS for execution, press End, or type CAN on the command line and press Enter.

JSUB returns to ISPF Edit on the JCL.

5. Edit the JCL as needed.
6. Repeat Steps 1 through 5 as needed for additional members.
7. Press End to terminate the ISPF Edit session.

#### JSUB Conditional Job Submission Panel

```

-----
COMMAND ==>> SUB          CONDITIONAL JOB SUBMISSION  ----- JOB/SCAN

    Do you wish to submit this job stream
    for execution?

    - To SUBMIT the scanned JCL even though it had errors
    type SUB in the COMMAND line and press <ENTER>.

    - To RETURN to the JOB/SCAN panel without submitting,
    type CAN in the COMMAND line and press <ENTER>.

    or

    - Press END.

```

## Validating JCL in Batch

The JOB/SCAN SJJSSAMP library contains two sample members, JSMVS and JS6PROC, that can be copied and modified to perform JCL Validation, in batch. This section describes the process of modifying the sample members and submitting a batch validation run. See [Reference Information](#) and [System Administrator Tasks](#) for detailed information about SJJSPARM library members, Input Selection statements, Report Options, Run Options, and Control Tables.

### Batch JOB/SCAN Overview

JOB/SCAN identifies the JCL to validate, determines which reports to generate and their formats, and establishes the run-time environment and controls, through Input Selection, Report Option, Run Option, and Control Table statements in ddname (file) JLRCL.

The JCL to run JOB/SCAN in batch provides the control statements to override and supplement the default controls. By contrast, when JOB/SCAN JCL Validation is invoked from the JOB/SCAN - JCL Validation panel, JLRCL is typically the concatenation of SJJSPARM library options members JSOPT01, JSOPT02, and JSOPT03, and a SYSIN file. The SYSIN file contains control statements determined by the settings of the JOB/SCAN - JCL Validation panel and the JOB/SCAN Processing Options panel fields.

There are two means of providing control for batch JOB/SCAN:

- Code the desired control statements in the SYSIN stream in the batch JOB/SCAN JCL.

If minimal control statement overrides are to be made, or if they will be used only once, code them in-stream in the batch JOB/SCAN JCL in the SYSIN stream. The options coded in control statements in the SYSIN stream override corresponding options coded in control statements in the SJJSPARM library members. This is the easiest means of identifying the input JCL dataset name and members to validate.

Figure - [JCL to run JOB/SCAN](#) shows an example of JOB/SCAN JCL with the input JCL dataset name and member name, control statement overrides, and reports to be generated contained in the JOB/SCAN JCL SYSIN stream.

Or

- Create an additional PARMLIB member containing the desired control statements.

If there are extensive control statement overrides or if they will be used repeatedly, code them in a new PARMLIB member, and code the batch JOB/SCAN JCL to pass the new member name to the JOB/SCAN PROC. Ask the JOB/SCAN Product Administrator to add the new PARMLIB member (`PRODOPT` in figure - [JCL to run JOBSCAN with different PARMLIB member](#)) to the SJJSPARM library.

Figure - [JCL to run JOBSCAN with different PARMLIB member](#) shows an example of JOB/SCAN JCL with the input JCL dataset name and member name contained in the JOB/SCAN JCL SYSIN stream and figure - [PARMLIB member PRODOPT contains](#) shows the control statement overrides and reports to be generated contained in the new PARMLIB member `PRODOPT`.

#### JCL to run JOB/SCAN

```
//JOBSCAN JOB (XXX,111),
//      'JOB/SCAN RUN',CLASS=A
//STEP01 EXEC JS6PROC,PRM=S
//SYSIN DD *
INPUT SELECTION
      DSN=DSSI.PROD.CNTL
      SELECT=PJGL0070
RUN OPTIONS
      CATALOG ACCESS=NO
REPORT OPTIONS
      STRUCTURED JCL LISTING=PRINT
```

#### JCL to run JOBSCAN with different PARMLIB member

```
//JOBSCAN JOB (XXX,111),
//      'JOB/SCAN RUN',CLASS=A
//STEP01 EXEC JS6PROC,PRM=S,
//      OPTON=PRODOPT
//SYSIN DD *
INPUT SELECTION
      DSN=DSSI.PROD.CNTL
      SELECT=PJGL0070
```

#### PARMLIB member PRODOPT contains

```
RUN OPTIONS
      CATALOG ACCESS=NO
REPORT OPTIONS
      OPTIONS IN EFFECT=SUMMARY
      STRUCTURED JCL LISTING=PRINT
```

**To execute JOB/SCAN JCL Validation in batch mode**

1. Make a copy of the SJSSAMP library member JSMVS.
2. Modify the JOB statement according to installation standards.

**SJSSAMP Library Member JSMVS**

```
//JSMVS    JOB (ACCT), 'PROGRAMMER NAME'
//*****
//* EXECUTE JOB/SCAN (MVS SYSTEMS)
//* REF:  SAMPLIB (JSMVS)
//*
//* NOTE: MODIFY JOB STATEMENT AND ITEMS INDICATED BY ARROWS <==
//*****
//JSSTEP1 EXEC JS6PROC, PRM=S
//SYSIN   DD   *
INPUT SELECTION
    DSN=YOUR.JCLLIB           <== INPUT JCL LIBRARY
    SELECT=YOURJOB           <== MEMBER TO SCAN
    SOURCE TYPE=PDS          <== TYPE OF INPUT LIB
REPORT OPTIONS
    OPTIONS IN EFFECT        =PRINT    <== PRINT OPE REPORT?
    STRUCTURED JCL LISTING=PRINT    <== PRINT SJL REPORT?
```

3. Modify the Input Selection detail statements as described in this table.

Detail Statement	Modification
<code>DSN=datasetname</code>	Identify the library containing the JCL to be validated.
<code>SELECT=memberselection</code>	Specify the members containing the JCL to be validated. Use masks, ranges and member names.
<code>SOURCE TYPE=</code>	Identify the type of library containing the JCL to be validated (that is, PDS, PANVALET, or LIBRARIAN).

4. Modify the Report Option statements to specify the required reports. Specify at least:

```
STRUCTURED JCL LISTING=PRINT
```

5. To override any Run Options coded in the SJSPARM library members, code a RUN OPTIONS group-level statement after the SYSIN DD statement. Follow the Run Options statement with the appropriate Run Option overrides. See [Run Option Statements](#) for more information about Run Option control statements.
6. Submit the JSMVS for batch execution. It executes JS6PROC.
7. View the SJL in the held output, particularly the JLKAC.



At least the summary version of the Options in Effect report should be printed to identify the Report Options and Run Options used for each run.

Control Table PARMLIB defaults cannot be overridden. JOB/SCAN uses the first table entry it encounters, and ignores duplicates. To cancel previous Control Table entries, remove them from the SJJSPARM library member in which they reside.

#### SJSSAMP Library Member JS6PROC

```

/*****
/** JOB/SCAN PROC - EXECUTED BY JSMVS (MVS SYSTEMS)
/** REF:  SAMPLIB(JS6PROC)
/**
/** NOTE: MODIFY ITEMS INDICATED BY ARROWS <==
/*****
//JS6PROC  PROC DJSNAME='XXXX.XXXX',          <== ASG PREFIX
//          OPT01=JSOPT01,
//          OPT02=JSOPT02,
//          OPT0N=JSOPT03,
//          DIAGOUT='SYSOUT=*',
//          SOUT='*',
//          PRM=S                               <== RUN TYPE
/*****
/** NOTE:  RUN TYPE 'S' = SCAN, 'C' = CHANGE, 'R' = REFORMAT,
/**        'CR' = CHANGE & REFORMAT, 'CS' = CHANGE & SCAN,
/**        'RS' = REFORMAT & SCAN, 'CRS' = CHANGE, REFORMAT, & SCAN.
/*****
//STEP01  EXEC PGM=J02YJS, PARM=&PRM, REGION=4096K
//STEPLIB DD DISP=SHR, DSN=&DJSNAME..SJJSLMOD
//*JLLOD  DD DISP=SHR, DSN=NAME.OF.LOADLIB (OPTIONAL EXIT LOAD LIB)
//JLRCL   DD DISP=SHR, DSN=&DJSNAME..SJJSPARM(&OPT01)
//        DD DISP=SHR, DSN=&DJSNAME..SJJSPARM(&OPT02)
//        DD DISP=SHR, DSN=&DJSNAME..SJJSPARM(&OPT0N)
//        DD DDNAME=SYSIN
//JLMSG   DD DISP=SHR, DSN=&DJSNAME..SJJSPARM
//JLCHLI  DD SYSOUT=&SOUT
//JLCHLO  DD SYSOUT=&SOUT
//JLSUMR  DD SYSOUT=&SOUT
//JLCHMO  DD DUMMY
//JLRULES DD DUMMY
//JLSSR   DD SYSOUT=&SOUT
//JLOPE   DD SYSOUT=&SOUT
//JLKAC   DD SYSOUT=&SOUT
//JLSNAP  DD &DIAGOUT
//JLDIAG  DD &DIAGOUT
//SYSUDUMP DD SYSOUT=&SOUT
//ABNLIGNR DD DUMMY                                IGNORE ABENDAID

```

See [Running Reformatter in Batch Mode](#) and [Running Reformatter in Batch Mode from Panels](#) for more information about other batch options.



# Advanced Topics for JCL Validation

This topic describes the advanced features of JOB/SCAN validation.

- [Selecting Members](#)
- [SJJSPARM Library Options Members](#)
- [Effects of Options on Validation Results](#)
- [Specifying a First Proclib for JCL Validation](#)
- [Searching a PROC](#)
- [Validating JCL that Includes CA-7 #J Statements](#)

## Selecting Members

The JOB/SCAN - JCL Validation panel fields, Select Members and Exclude Members, provide a flexible means of selecting members of the dataset specified in the Input Library field.

### Selecting Members By Name from the Input Library Dataset

Member names can be specified, in either the Select Members field or the Exclude Members field, with any combination of these options:

- One or more specific member Names.
- Member names included within a specified sort Range.
- Member names beginning with the same characters (Prefix).
- Member names that fit a mask of matching and wildcard characters (\*).
- All of the member names.

This section includes an example of each of the selection options. See [Selection Option Descriptions](#) for complete details about using these options to select JCL members for JOB/SCAN processing.

### *To Select Members by Name from the Input Library Dataset*

1. Specify the input library dataset name in the Input Library field (see figure - [JOB/SCAN - JCL Validation Panel showing Member Selection](#)).
2. Specify the input type (J for JOBS or P for PROCs) in the Input Type field, and the dataset organization (PDS for partitioned organization) in the Library Type field (see figure - [JOB/SCAN - JCL Validation Panel showing Member Selection](#)).
3. In the Select Members field, type as many entries, separated by commas, as are needed to select all of the members to process. Use any combination of the selection options described in [Selection Option Descriptions](#).

### Entering the Options

The selection options Range, Prefix, Mask, and ALL can select unwanted members along with needed ones. These unwanted members can be excluded from the validation run by specifying their names in the Exclude Members field. All of the selection options that are valid in the Select Members field are also valid in the Exclude Members field.

The scope of selection options Range, Prefix, Mask, and ALL can overlap or include members already specified by Specific Name, but will not result in repeating the selection of any member from any of the selection options occurring earlier in the list.

Members that have been selected by Specific Name in the Select Members field cannot be excluded by entries in the Exclude Members field. Members that have been selected by Range, Prefix, Mask, or ALL selection options in the Select Members field can be excluded by entries in the Exclude Members field.

#### JOB/SCAN - JCL Validation Panel showing Member Selection

```

----- JOB/SCAN - JCL VALIDATION ----- PROCESS COMPLETE
COMMAND ==> F

      F - Foreground execution mode      P - Processing Options
      B - Batch execution mode           E - Edit JCL member
      S - Submit After Validation        BR - Browse JCL member
Input Library ==> 'ASGCOM2.BOOK.JOBS'
Input Type   ==> J                      (J - JOB JCL, P - PROC, or L - Selection List)
Library Type ==> PDS                    (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> MYJOB,MYJOB01-MYJOB10,YOURJOB*
Exclude Members ==> MYJOB03
First Proclib ==>
Proclib Type ==>                      (PDS, PAN, LIB, or LIB-3)
...

```

In the Select Members field, type as many entries, separated by commas, as are needed to select all of the members to process. Use any combination of the selection options described [Selection Option Descriptions](#). The 54-byte input field length is the only limitation to the number of entries. The members will be validated in the order in which they are entered. Member names can be repeated (but only by Specific Name) and will be validated as many times as they are entered. Do not use blanks between entries; a blank in the input field will terminate the list.

The member names resulting from a selection option entry of Range, Prefix, Mask, or ALL will be processed in sort order sequence within the set resulting from that selection option entry. But, the *set* of members so selected will be processed in the order *entered* relative to members selected by other selection option entries.

### Selection Option Descriptions

This section describes the selection options for the Select Members and Exclude Members fields.

#### *Specific Name*

Type in any number of specific member names, separated by commas (no blanks permitted), in any order. Some examples:

```

Select Members ==> JOB003
Select Members ==> JOB003, JOB729, JOB018, JOB729

```

Members of the input library dataset specifically named in the Select Members field cannot be excluded by entries in the Exclude Members field. They can be specified for processing any number of times in the same JCL Validation run, and will be processed in the order in which they are entered in the Select Members field. Specific member names in the Exclude Members field exclude members that would otherwise be included in the JCL Validation run by Range, Prefix, Mask, or ALL options entered in the Select Members field.

## Range

Type two member names, separated by a hyphen, to define the range of member names to select (in the fields Select Members and Exclude Members). For example:

```
Select Members   ==> JOB000-JOB008
Exclude Members ==> JOB004-JOB006
```

The names do not need to match actual member names, but must be entered in sort order. The Range selects (or excludes) members whose names match either of the entered names or fall between the two in sort order. The Range does not duplicate any member name from any of the selection options occurring earlier in the list. For example, using the ranges specified in this section, if the input library dataset contains the members JOB001, JOB002, JOB003, JOB004, JOB005, JOB006, JOB007, and JOB008, then, JOB/SCAN validates JOB001, JOB002, JOB003, JOB007, and JOB008 in that order.

## Prefix

Type PREFIX= followed by characters matching the first characters of the names of members to be selected. For example:

```
Select Members   ==> PREFIX=JOB0
```

This example will result in selecting those members whose names begin with the character string JOB0, no matter what the remaining characters of their names. Prefix will not result in duplicating any member name from any of the selection options occurring earlier in the list.

## Mask

Type asterisks as wildcard characters in the mask positions corresponding to character positions to be ignored. Type the actual characters to be matched, in the character positions to be matched, in the member names to be selected. For example:

```
Select Members   ==> **B0*3**
```

This example will select members of the input library dataset with member names whose third, fourth, and sixth characters are B, 0, and 3, respectively. The length of the mask must be equal to or greater than the length of the name of any member to be selected. Mask will not result in duplicating any member name from any of the selection options occurring earlier in the list.

## ALL

To select *all* of the members of the input library dataset, in sort order sequence, type *ALL* in the Select Members field. For example:

```
Select Members   ==> ALL
```

## Combination

You may combine several of the selection options, for example:

```
Select Members   ==> MYJCL1,MYJCL3-MYJCL5,ALL
```

## Exclude Member

You may exclude members from processing only when they have been selected by Range, Prefix, Mask or ALL. Members selected by whole names may not be excluded.



- A member name of 'ALL' will cause JOB/SCAN to load each member name in the PDS, regardless of any specified count.
- The 54-byte input field length is the only limitation to the number of entries.
- The members will be validated in the order in which they are entered.
- Do not use blanks between entries; a blank in the input field terminates the list.
- Member names can be repeated (but only by Specific Name) and will be validated as many times as they are entered.
- If you want JOB/SCAN to automatically prepend your user ID as a high-level-qualifier, type unquoted values into the Input Library and First Proclib fields. Otherwise, place quotes around the library name to fully specify a library and JOB/SCAN uses them as-is.
- ALL will not result in duplicating any member name from any of the selection options occurring earlier in the list.

## Selecting Members from the Input Library JCL Directory

The JCL Directory panel allows selection from a list of all members of the dataset specified in the JOB/SCAN - JCL Validation panel's Input Library field.

### To access the JCL Directory panel

1. On the JOB/SCAN - JCL Validation panel, specify the execution mode (F for foreground or B for batch) in the Command field.

```

----- JOB/SCAN - JCL VALIDATION -----
COMMAND ==> F

      F - Foreground execution mode          P - Processing Options
      B - Batch execution mode              E - Edit JCL member
      S - Submit After Validation           BR - Browse JCL member
Input Library  ==> 'ASGCOM2.BOOK.JOBS'
Input Type    ==> J      (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ==> PDS   (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==>
Exclude Members ==>
First Proclib ==>
Proclib Type  ==>      (PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
Report Heading ==>
Display        ==> Y (Yes/No/ErrorsOnly)  Browse/View ==> B (B/V)
Hardcopy       ==> N (Yes/No/ErrorsOnly/SpaceOnly)
Options in Effect Listing ==> N (Y/N)      SJL PDSOUT ==> N (Y/N)

Selection Exit ==>      Parm ==>
Standards Name ==>      Parm ==>

```

2. Specify the input library dataset name in the Input Library field.
3. Specify the input type (J for JOBS or P for PROCs) in the Input Type field, and the dataset organization (PDS for partitioned organization) in the Library Type field.
4. Leave the Select Members field blank and press Enter. JOB/SCAN ignores the Exclude Members field if the Select Members field is blank.

The JCL Directory panel will appear, displaying a list of the members contained in the input library, similar to this figure. Members can be selected, browsed, or edited at this point.

```

----- JCL DIRECTORY ----- ROW 1 TO 9 OF 9
COMMAND ==> (Use DISPLAY or CANCEL if needed) SCROLL ==> PAGE

Library: BOOK.JOBS

Place B or E next to members to Browse or Edit. Press ENTER.
Place S next to members to Select. Press Enter. Press END to process.

Cmd  Member      Seq
     Name        Nbr

     JOB001
     JOB002
     JOB003
     JOB004
     JOB005
     JOB006
     JOB007

```

The JCL Directory panel processing allows selecting members of the input library dataset in their normal sort order or selecting them in a specific sequence.

5. Choose whether processing is by directory (sorted) order or by a selected sequence.

#### ***To select members in the order in which they appear in the directory***

Use this procedure to select several members when the order of validation is the order in the list.

- a. Select all the members to be processed by typing *S* in the Cmd field to the left of the name of each member to be selected.
- b. Press Enter.

```

----- JCL DIRECTORY ----- ROW 1 TO 9 OF 9
COMMAND ==> (Use DISPLAY or CANCEL if needed) SCROLL ==> PAGE

Library: BOOK.JOBS

Place B or E next to members to Browse or Edit. Press ENTER.
Place S next to members to Select. Press Enter. Press END to process.

Cmd  Member      Seq
     Name        Nbr

     JOB001
S    JOB002
     JOB003
S    JOB004
S    JOB005
S    JOB006
     JOB007

```

The JCL Directory panel displays numbers in the Seq Nbr field to the right of the names of the selected members. These numbers indicate the order in which the selected members will be processed by JOB/SCAN JCL Validation.

```

----- JCL DIRECTORY ----- ROW 1 TO 9 OF 9
COMMAND ==> (Use DISplay or CANcel if needed) SCROLL ==> PAGE

Library: BOOK.JOBS

Place B or E next to members to Browse or Edit. Press ENTER.
Place S next to members to Select. Press Enter. Press END to process.

Cmd  Member      Seq
     Name        Nbr

     JOB001
     JOB002        1
     JOB003
     JOB004        2
     JOB005        3
     JOB006        4
     JOB007

```

### To set the validation sequence for selected members

Use this procedure to select several members when the order of validation is not the order in the list.

- a. Type S in the Cmd field to the left of the name of the first member to be validated.

```

... Place S next to members to Select. Press Enter. Press END to process.

Cmd  Member      Seq
     Name        Nbr

S    JOB001
     JOB002

```

- b. Press Enter. This figure shows the sequence number on the selected member.

```

... Place S next to members to Select. Press Enter. Press END to process.

Cmd  Member      Seq
     Name        Nbr

     JOB001        1
     JOB002

```

- c. Type S in the Cmd field to the left of the name of the next member to be validated.

```

... Place S next to members to Select. Press Enter. Press END to process.

Cmd  Member      Seq
     Name        Nbr

     JOB001        1
... S    JOB004

```

d. Press Enter. This figure shows the next number on the selected number.

```

... Place S next to members to Select. Press Enter. Press END to process.

Cmd  Member      Seq
     Name        Nbr
...  JOB001        1
     JOB004        2

```

e. Repeat steps c and d for each member to be processed, selecting the members in the order in which they are to be processed.

```

----- JCL DIRECTORY ----- ROW 1 TO 9 OF 9
COMMAND ==> (Use DISPLAY or CANCEL if needed) SCROLL ==> PAGE

Library: BOOK.JOBS

Place B or E next to members to Browse or Edit. Press ENTER.
Place S next to members to Select. Press Enter. Press END to process.

Cmd  Member      Seq
     Name        Nbr
     JOB001        1
     JOB002        4
     JOB003
     JOB004        2
     JOB005        3
     JOB006
     JOB007        5

```

6. Choose one of these options:

- Press End to invoke JOB/SCAN JCL Validation to process the members.

**Or**

- Enter the DISPLAY command on the command line to display a list of the selected members in the order in which they will be validated. This display shows the result of Step 5e.

```

----- JCL DIRECTORY ----- ROW 1 TO 9 OF 9
COMMAND ==> (Use DISPLAY or CANCEL if needed) SCROLL ==> PAGE

Library: BOOK.JOBS

Place B or E next to members to Browse or Edit. Press ENTER.
Place S next to members to Select. Press Enter. Press END to process.

Cmd  Member      Seq
     Name        Nbr
     JOB001        1
     JOB004        2
     JOB005        3
     JOB002        4
     JOB007        5

```

**Or**

- Enter the CANCEL command to stop processing and return to the JOB/SCAN - JCL Validation panel.

This table lists the primary commands for the JCL Directory panel command line. (Letters in parentheses in the commands are optional.)

Primary Command	Description
CAN(CEL)	Stops the selection process and returns to the JOB/SCAN - JCL Validation panel.
L(OCATE) <i>membername</i>	Followed by the name of a member in the directory - will locate that member name and move the cursor to the Cmd field to the left of the member name.
DIS(PLAY)	Lists the names of the selected members in the order in which JOB/SCAN. processes them. Selections cannot be changed after displaying the list. If the list is not correct, use the CANCEL command to begin again.

### **To browse or edit a member of the input library dataset**

1. On the JCL Directory panel, type B or E (for browse or edit) in the Cmd field to the left of the member name.
2. Press Enter.

JOB/SCAN displays the member in ISPF browse or edit mode.

3. To exit Browse or Edit, press End.

### **Selecting Jobs with JOB/SCAN Selection Lists**

The List Library Directory panel and the Modify Selection List panel allow selection of jobs or prepared lists of jobs from one or more JOB/SCAN Selection Lists.

JOB/SCAN Selection Lists provide a means of grouping jobs that are frequently validated together, in the order in which they are to be validated. After creating a Selection List, use it to repeatedly validate the group of jobs without needing to redo a lengthy selection process. See [Creating a JOB/SCAN Selection List](#) for information about creating Selection Lists.

### **To select jobs for validation from selection lists**

1. On the JOB/SCAN - JCL Validation panel, specify the execution mode (F for foreground or B for batch) in the Command field.
2. Type the Selection List library name in the Input Library field.
3. Type  $\perp$  in the Input Type field.
4. Type PDS in the Library Type field.
5. To select a single Selection List library member by name, type the member name in the Select Members field, press Enter, and skip to Step 7.
6. To select any number of members from the List Library directory, leave the Select Members field blank, and press Enter.

JOB/SCAN displays the List Library Directory panel, showing a list of members contained in the Selection List library.

7. At this point, select the Selection List members to be used for the JCL Validation run.



The members can be selected in any order by typing *s* in the Option field and pressing Enter for each member, one at a time.

8. When all desired members have been selected, press End.

The Modify Selection List panel appears for each selected Selection List member. At this point, the contents of the Selection List member can be temporarily modified (for this JCL Validation run only; the actual Selection List member is not changed). See the online help (PF1) for more information about the fields of this panel.

9. Make any desired modifications to the Selection List. When the modifications are complete, press End to proceed.

10. Repeat Step 7 for each selected Selection List member.

#### JOB/SCAN - JCL Validation Panel: Selecting Jobs with JOB/SCAN Selection Lists

```

----- JOB/SCAN - JCL VALIDATION -----
COMMAND ==> F

      F - Foreground execution mode          P - Processing Options
      B - Batch execution mode              E - Edit JCL member
      S - Submit After Validation           BR - Browse JCL member
Input Library  ==> 'DSSI.#COM2.LIST'
Input Type    ==> L                        (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ==> PDS                      (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> LIST0A
Exclude Members ==>
First Proclib ==>
Proclib Type  ==>                          (PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
Report Heading ==>
Display        ==> Y (Yes/No/ErrorsOnly)  Browse/View ==> B (B/V)
Hardcopy       ==> N (Yes/No/ErrorsOnly/SpaceOnly)
Options in Effect Listing ==> N (Y/N)      SJL PDSOUT ==> N (Y/N)

Selection Exit ==>                          Parm ==>
Standards Name ==>                          Parm ==>

```

#### List Library Directory Panel: Showing Members LIST1A and LIST0A Selected in Reverse Sequence

```

----- LIST LIBRARY DIRECTORY ----- Row 1 to 6 of 6
COMMAND ==>                               SCROLL ==> PAGE

DSNAME: 'DSSI.#COM2.LIST'

Place (S)elect or (B)rowse next to LIST(S) required and press <ENTER>.
When all required LISTS are numbered, press <END> to process.
Use (CAN)cel in the Command Line to abandon further processing.

OPTION  NAME
  S     LIST0A      2
       LIST1A      1
       MFG
       PAYROLL
       PROD
       TEST

```

**Modify Selection List Panel**

```

MODIFY SELECTION LIST - LIST0A ----- Row 1 to 9 of 9
COMMAND ==>                               SCROLL ==> PAGE

                (Use END to save and continue,
                ENTER to process changes,
                CANCEL to abandon processing)

JCL Library      = BOOK.JOBS
Library Type     = PDS
List Description = SELECTION LIST

      Jobname    (Use D, C, B, E, I, M, A, or P )
      JOB001
      JOB002
B     JOB003
      JOB004
      JOB005
      JOB006
      JOB007
      JOB008
      JOB009

```



For help with the commands of the Modify Selection List panel, put the cursor on the Command field and press Help (PF1).

**Creating a JOB/SCAN Selection List**

JOB/SCAN Selection Lists are specially formatted members stored in partitioned datasets designated as *List Libraries*. A Selection List identifies a set of jobs that are frequently processed together by JOB/SCAN. Information in the Selection List includes a brief description of the list, the dataset name and dataset organization of the JCL library containing the set of jobs, and Input Selection control statements to select all members of the set of jobs in the Selection List.

**Selection List Creation Procedure**

The List Library which will contain the JOB/SCAN Selection List must be an existing partitioned dataset. (If the List Library does not already exist, use the ISPF utility functions to create one.)

**To create a new JOB/SCAN Selection List:**

1. Display the Selection Lists Maintenance Facility panel.
  - a. Choose the Selection List Maintenance option from the ASG-JOB/SCAN Primary Menu panel.

For example:

**Selection Lists Maintenance Facility Panel: CREATE Option**

```

----- SELECTION LISTS MAINTENANCE FACILITY -----
COMMAND ===> c

          C - Create a Selection LIST
          M - Modify a Selection LIST

LIST Library    ===> 'DSSI.#COM2.LIST'

LIST Name       ===> list0a      (Required if Option "C" selected)
                                   (Leave blank for LIST Directory)

Use ISPF UTILITIES functions to Delete, Copy, Print, and Rename LISTS

```

- b. On the Selection Lists Maintenance Facility panel, on the command line type C to create a new Selection List.
- c. In the LIST Library field, type the dataset name of the List Library that is to contain the new Selection List.
- d. In the LIST Name field, type the name of the new Selection List.
- e. This must be a valid member name, unique in this List Library dataset.
- f. Press Enter. JOB/SCAN displays the Create a Selection List panel.

**Create a Selection List Panel**

```

----- CREATE A SELECTION LIST -----
COMMAND===>

List Name       : LIST0A
List Description ===> documentation test list

JCL Library     ===> 'DSSI.BOOK.JOBS'
Library Type    ===> pds

          Select Criteria (optional)
Range:  Start ===> job000      End ===> job010
Prefix/Mask    ===>

JCL Library and Library Type must be specified.

Press ENTER to get the Directory of JCL Members contained in the JCL
Library.

```

2. Set up the meta-characteristics of the Selection list.

The Create a Selection List panel establishes the identity of the new Selection List, identifies the associated JCL Library, and begins the process of selecting and ordering the job member names to be included in the Selection List.

- a. (Optional) In the List Description field, type a brief description (up to 32 characters) of the new Selection List.
- b. In the JCL Library field, type the dataset name of the library containing the JCL members that are to be grouped by this Selection List.
- c. In the Library Type field, indicate the type of JCL library. Values accepted in this field are:
  - PDS (for partitioned dataset)
  - PROCLIB
  - PANVALET (or PAN)

LIBRARIAN (or LIB)

LIBRARIAN-3 (or LIB-3)

- d. (Optional) Specify selection criteria to limit the member names displayed for selection in the Contents of (listname) panel. You can specify one of these:

a *Range* of member names, a *Prefix*, a *Mask*.

For more information about using selection criteria, place the cursor in the appropriate input field and press <Help>.

The Contents of (listname) panel will display the entire directory list of the JCL Library for selection if selection criteria are not specified.

- e. Press Enter; JOB/SCAN displays the Contents of (listname) panel.

#### Contents of (listname) Panel

```

----- CONTENTS OF LIST0A ----- ROW 1 TO 9 OF 9
COMMAND ==> SCROLL ==> PAGE

      (Enter END to continue, CANCEL to return without creating)

List Name       : LIST0A
JCL Library    : 'DSSI.BOOK.JOBS'
Library Type   : PDS
List Description ==> DOCUMENTATION TEST LIST

Cmd  Member      Seq  (Use S(elect), B(rowse), or E(dit) in the Cmd column)
s    JOB001
      JOB002
s    JOB003
      JOB004
s    JOB005
      JOB006
s    JOB007
...

```

3. Initialize the order of the Selection list.

This panel continues the process of selecting and ordering job members for inclusion in the new Selection List, and provides an opportunity to alter the List Description.

The Contents of (listname) panel displays a list of the job member names remaining for the new Selection List after applying selection criteria on the Create a Selection List panel. If no selection criteria were entered on the Create a Selection List panel, then the list is a list of all members in the JCL Library dataset.

#### **To browse or edit a member of the Selection List**

- Type B or E in the Cmd field to the left of the member name and press Enter.

If multiple B or E commands are entered at the same time, the members are displayed consecutively.

**To include members in the Selection List**

- a. Type `s` in the `Cmd` field to the left of each member to be included.
- b. Press `Enter`. Sequence numbers indicating the order in which the members will be processed by `JOB/SCAN` appear to the right of the member names.
- c. To specify the ordering, select the members one at a time in the desired order, pressing `Enter` for each selection. Otherwise, the members will be processed in the order they are displayed (top to bottom).
- d. When member selection is complete, press `End`; `JOB/SCAN` displays the `Modify List` panel.

## 4. Finalize the order of the Selection list.

This panel displays the list of job member names that has been selected and ordered by the previous panels. It provides the facility for final editing of the list.

**Modify List Panel**

```

MODIFY LIST - LIST0A      -----  ROW 1 TO 5 OF 5
COMMAND ==>              SCROLL ==> PAGE

(Enter END to save, ENTER to process, CANCEL to quit without saving)

JCL Library      ==> 'DSSI.BOOK.JOBS'
Library Type     ==> PDS
List Description ==> DOCUMENTATION TEST LIST

Jobname         (Use D, C, B, E, I, M, A, or P )
JOB001
JOB003
JOB007
JOB005

```

This table lists primary commands you may enter on the command line of the `Modify List` panel when modifying the Selection List:

Command	Description
CANCEL	Return to the previous panel without saving changes.
IFIRST	Insert a member name at the beginning of the list.
RESET	Remove a line command that was entered inadvertently.

This table lists line commands available to Edit the list of members:

Command	Description
B	Browse this member.
E	Edit this member.

Command	Description
D	Delete this member name from the list.
C	Change this member name.
I	Insert a new member name after this line.
M	Move this line.
A	After this line - establishes target for M (Move) line command.
P	Prior to this line - establishes target for M (Move) line command.

Use these primary and line commands to delete, change, insert, or reorder the job member names for the Selection List. Ensure that the list is correct, then press End to save the list and return to the Selection Lists Maintenance Facility panel. The new Selection List has now been created and can be used whenever this job stream is to be validated.

### Modifying a JOB/SCAN Selection List

JOB/SCAN Selection Lists are specially formatted members stored in partitioned datasets designated as *List Libraries*. A Selection List identifies a set of jobs that are frequently processed together by JOB/SCAN. Information in the Selection List includes a brief description of the list, the dataset name and dataset organization of the JCL library containing the set of jobs, and Input Selection control statements to select all members of the set of jobs in the Selection List.

Two JOB/SCAN panels are used in the order shown to modify an existing JOB/SCAN Selection List:

- Selection Lists Maintenance Facility panel — to specify the MODIFY option, the List Library dataset that contains the Selection List to be modified, and the member name of the Selection List to be modified.
- Modify List panel — for editing and ordering the job member names in the Selection List.

#### **To modify an existing JOB/SCAN selection list**

1. Access the Selection Lists Maintenance Facility panel by choosing the Selection List Maintenance option from the ASG-JOB/SCAN Primary Menu panel.
2. On the command line, type M for Modify; this is an existing Selection List.
3. In the LIST Library field, type the dataset name of the List Library that contains the Selection List to be modified.
4. In the LIST Name field:
  - type the name of the Selection List to be modified. This must be the member name of a member in this List Library dataset.

**Or**

- Leave this field blank and press Enter to display the List Library directory and select the member to be modified.

#### Selection Lists Maintenance Facility Panel: MODIFY Option

```

----- SELECTION LISTS MAINTENANCE FACILITY -----
COMMAND ===> m

          C - Create a Selection LIST
          M - Modify a Selection LIST

LIST Library    ===> 'DSSI.#COM2.LIST'

LIST Name      ===> list0a      (Required if Option "C" selected)
                               (Leave blank for LIST Directory)

Use ISPF UTILITIES functions to Delete, Copy, Print, and Rename LISTS

```

5. Press Enter to proceed to the Modify List panel.

## SJJSPARM Library Options Members

The SJJSPARM library contains options members that are read during JOB/SCAN execution. These options members contain Access Control, Run Options, Report Options, Control Tables, and File Allocations statements that:

- Tailor JOB/SCAN ISPF panels for users.
- Set up the JOB/SCAN run-time environment.
- Determine report content and formats.
- Specify file allocation information.

Administrators keep the members of the SJJSPARM library up to date.

### Using a Personal Options Member

The JOB/SCAN administrator creates and maintains personal options members. Ask your administrator to create or change your options member and test it. You determine when to use the personal options member.

#### ***To use the personal options member instead of JSOPT02 or JSOPT03***

1. Select Assign Alternate Options Members from the ASG-JOB/SCAN Primary Menu panel.
2. Type the name of the new options member in the Personal Initialization field if it is to be used instead of JSOPT02, or in the Personal Override field if it is to be used instead of JSOPT03.
3. Press End to verify the existence of the personal options member and return to the ASG-JOB/SCAN Primary Menu panel.

**Assign Alternate Options Members Panel**

```

----- ASSIGN ALTERNATE OPTIONS MEMBERS -----
COMMAND ===>

CHANGE THE OPTIONS MEMBERS AS NEEDED:

      Initialization      Personal      Assigned      Default
      Override            ===> PRSOPTP2  ===>          ===> JSOPT02
                        ===> PRSOPTP3  ===>          ===> JSOPT03

Press ENTER to verify the existence of Personal and Assigned Options Members
Enter END/PF3 to save options members with verification.

```

4. Exit and restart JOB/SCAN.

**Default SJJSPARM Library Options Members**

JOB/SCAN concatenates the SJJSPARM library options members that it uses for JCL Validation in the ddname JLRCL. The default options members are listed in this table:

Member	Description
JSOPT01	Contains control information.
JSOPT02	Contains the File Allocations control statements required for initialization when the Validate JCL option is chosen from the ASG-JOB/SCAN Primary Menu panel. Typically contains most of the File Allocations, Run Options, and Report Options statements used when JOB/SCAN is invoked from the JOB/SCAN - JCL Validation panel and when JOB/SCAN Edit macros are used. Can also contain user-supplied Control Table statements for the Reformatter and the Standards Facility.
JSOPT03	Contains additional control statements used when JOB/SCAN JCL Validation is invoked from the JOB/SCAN - JCL Validation panel and when JOB/SCAN Edit macros are used. JSOPT03 is concatenated to ddname JLRCL following JSOPT02, and will override conflicting statements in JSOPT02.
JSOPT04	Contains Input Selection, Run Options, and Report Options statements used by JOB/SCAN Edit macros. Can also be used to specify the Standards Facility to be used with the Edit macros.
JSOPT05	JSOPT05 is used only when JCL Validation is invoked by JOB/SCAN Edit macros. JSOPT05 is concatenated to ddname JLRCL following JSOPT02 and JSOPT03, and will override conflicting statements in JSOPT02 and JSOPT03.

In addition to these SJJSPARM library options members, JOB/SCAN also uses control information entered from the JOB/SCAN Processing Options panel and the JOB/SCAN - JCL Validation panel to establish the options used when JCL Validation is invoked from the JOB/SCAN - JCL Validation panel.





- The SJJSPARM library options members described in [Default SJJSPARM Library Options Members](#) are the default options members. JOB/SCAN provides the means of assigning alternate options members to use instead of JSOPT02 or JSOPT03. (See [Managing JOB/SCAN through Options Members](#) for more information about alternate options members.)
- JOB/SCAN can also create a JSOPTJ2 options member to use, in addition to the other options members, to provide default File Allocations statements for ddname JLJPA. (See [File Allocations Group](#) for more information about creating default File Allocations statements.)
- Option in personal options members are used before those in default or assigned members.

### Processing SJJSPARM Library Options Members

This section describes how JOB/SCAN processes SJJSPARM library options members for JCL Validation in the foreground.

### Options Member Search Order

When a user invokes JOB/SCAN from the ISPF Primary Option Menu panel, JOB/SCAN reads member JSOPT01 to determine the JOB/SCAN facilities the user is authorized to access. The Access Control statements in JSOPT01 determine the configuration of the user's ASG-JOB/SCAN Primary Menu panel. When JCL Validation is invoked, the SJJSPARM library options members are allocated to ddname JLRCL, concatenated and read in the order shown in the table in [Default SJJSPARM Library Options Members](#).

After scanning the SJJSPARM library options members, if the JCL Validation is invoked from the JOB/SCAN - JCL Validation panel, JOB/SCAN JCL Validation scans the options entered from the JOB/SCAN Processing Options panel and the JOB/SCAN - JCL Validation panel. The options entered from these panels override any conflicting options read from the SJJSPARM options members.

### Control Statement Precedence

JOB/SCAN gives precedence to the *last* control statement encountered for a given option. That is, an option control statement will override a conflicting option control statement read *earlier* in the search, except that statements in personal options members take precedence.



An exception to this general rule is the selection precedence for Control Table entries. The *first* value encountered for a Control Table entry takes precedence over subsequent, conflicting values for the same Control Table entry.

### Precedence of SJJSPARM Options Members

JOB/SCAN searches for option control statements and panel options in this order, when invoked from the JOB/SCAN - JCL Validation panel or Edit macro:

1. JSOPT01
2. (Optional) Initialization personal options member
3. JSOPT02
4. (Optional) Override personal options member

5. JSOPT03
6. JSOPTM (if JCL Validation is invoked from JOB/SCAN Edit macros)
7. JOB/SCAN Processing Options panel (if JCL Validation is invoked from the JOB/SCAN - JCL Validation panel)
8. JOB/SCAN - JCL Validation panel (if JCL Validation is invoked from the JOB/SCAN - JCL Validation panel)

### Relating JCL Validation Panel Fields to SJJSPARM Options

Report Option and Run Option parameters from the JOB/SCAN - JCL Validation panel and the JOB/SCAN Processing Options panel override the corresponding parameters from SJJSPARM library options members when running JOB/SCAN from the JOB/SCAN - JCL Validation panel. These tables show the correspondence between JOB/SCAN - JCL Validation panel fields and SJJSPARM option control statement parameters. (Default values appear underlined.) See [Relating Processing Options Panel Fields to SJJSPARM Options](#) for similar information about JOB/SCAN Processing Options panel fields.

Panel Field	SJJSPARM Statement and Parameter
DISPLAY	STRUCTURED JCL DISPLAY = <u>YES</u>   NO   ERRORONLY
BROWSE/VIEW	STRUCTURED JCL DISPLAY = <u>BROWSE</u>   VIEW
HARDCOPY	STRUCTURED JCL LISTING =YES   <u>NO</u>   ERRORONLY   SPACEONLY   JOBERRORS
OPTIONS IN EFFECT LISTING	OPTIONS IN EFFECT =YES   <u>NO</u>

Panel Field	SJJSPARM Statement and Parameter
LIBRARY TYPE	JLJPA PROCLIB TYPE = <u>PDS</u>   PAN   LIB   LIB-3   SEQ
PROCLIB TYPE	JLJPC PROCLIB TYPE = <u>PDS</u>   PAN   LIB   LIB-3
REPORT HEADING	REPORT HEADING =' <i>report heading</i> '
STANDARDS NAME AND PARM	STANDARDS NAME = <i>prgname</i> , ' <i>parm</i> '

## Sample JOB/SCAN - JCL Validation Panel

```

----- JOB/SCAN - JCL VALIDATION -----
COMMAND ==> F

      F - Foreground execution mode      P - Processing Options
      B - Batch execution mode           E - Edit JCL member
      S - Submit After Validation        BR - Browse JCL member
Input Library  ==> BOOK.TESTLIB
Input Type    ==> J                      (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ==> PDS                     (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> PUBTEST
Exclude Members ==>
First Proclib ==>
Proclib Type  ==>                        (PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
Report Heading ==> JCL VALIDATION PUBLICATION
Display        ==> Y (Yes/No/ErrorsOnly)  Browse/View ==> B (B/V)
Hardcopy       ==> N (Yes/No/ErrorsOnly/SpaceOnly)
Options in Effect Listing ==> N (Y/N)      SJL PDSOUT ==> N (Y/N)

Selection Exit ==>                        Parm ==>
Standards Name ==>                        Parm ==>

```

## Relating Processing Options Panel Fields to SJJSPARM Options

Report Option and Run Option parameters from the JOB/SCAN - JCL Validation panel and the JOB/SCAN Processing Options panel override the corresponding parameters from SJJSPARM library options members when running JOB/SCAN from the JOB/SCAN - JCL Validation panel. This table shows the correspondences between JOB/SCAN Processing Options panel fields and SJJSPARM library option control statement parameters. (Default values are **BOLD**.) See the tables in [Relating JCL Validation Panel Fields to SJJSPARM Options](#) for corresponding information about JOB/SCAN - JCL Validation panel fields.

Panel Field	Option Type	SJJSPARM Statement and Parameter
Use Error Table	Run Option	USE ERROR TABLE = <b>YES/NO</b>
Condition Code Processing	Run Option	PROCESSING SPECIFICATIONS =CONDCODE/ <b>NOCONDCODE</b>
Catalog Access	Run Option	CATALOG ACCESS = <b>YES/NO</b>
VTOC Access	Run Option	CATALOG ACCESS = <b>VTOC/NOVTOC</b>
PDS Member Verification	Run Option	CATALOG ACCESS = <b>PDSMEM/NOPDSMEM</b>
Security Exit Dsname Check	Run Option	SECURITY EXIT =,DSNAME/ <b>NODSNAME</b>
Program Checking	Run Option	CATALOG ACCESS = <b>PGM/NOPGM</b>
Disable SMS Processing	Run Option	SMS =ACS/ <b>NOACS,CAS/NOCAS</b>
IDCAMS Processing	Run Option	PROGRAM PRODUCT INTERFACE =IDCAMS/ <b>NOIDCAMS</b>

Panel Field	Option Type	SJJSPARM Statement and Parameter
Job Limit	Run Option	JOB LIMIT =(,y) where y = 0-99998 and 0 = no limit.
Access Control Cards	Run Option	ACCESS CONTROL CARDS =YES/NO
Control Card Limit	Run Option	ACCESS CONTROL CARDS =LIMIT=n
Subsystem	Run Option	SUBSYSTEM =JES2/JES3/NONE
Suppress Error Messages	Run Option	SUPPRESS ERROR MESSAGES =NO/A/W/E/nn where nn = 0 - 99 is the Severity Level.
Extended Error Text	Report Option	STRUCTURED JCL LISTING=ERRORTXT=SHORT/EXTENDED



- The value in the Job Limit field overrides the value in the JOB LIMIT statement from the SJJSPARM library, i.e. the *y* in JOB LIMIT =(, *y*), only if it is less than *y*.
- You may code values for Access Control Card and Control Card Limit in one statement:  
Access Control Cards=YES, LIMIT=nn

#### Sample JOB/SCAN Processing Options Panel

```

----- JOB/SCAN PROCESSING OPTIONS -----
COMMAND ===>

Use Error Table           ===> Y           (Y/N)
Condition Code Processing ===> Y           (Y/N)
Catalog Access           ===> Y           (Y/N)
VTOC Access              ===> Y           (Y/N)
PDS Member Verification  ===> Y           (Y/N)
Security Exit DSNNAME Check ===> Y       (Y/N)
Program Checking         ===> Y           (Y/N)
Disable SMS Processing   ===> Y           (Y/N)
IDCAMS Processing        ===> Y           (Y/N)
Use SCHEDULER interface  ===> Y           (Y/N)
Job Limit                ===>           (1 to 999 or Blank For No Limit)
Access Control Cards      ===> Y           (Y/N)
Control Card Limit       ===>           (1 to 999 or Blank For No Limit)
Subsystem                ===> JES2       (JES2/JES3/None)
Suppress Error Messages  ===> NO        (A,W,E,0 to 99,or NO)
Extended Error Text      ===> S           (X=Extended, S=Short)

Press <ENTER> to verify option values or <END> to leave and save option values.

```

## Effects of Options on Validation Results

JOB/SCAN can use the same SJJSPARM options members (usually JSOPT01, JSOPT02, and JSOPT03) to specify options and controls for its three modes of operation. The modes of operation (panels, Edit macros, and batch) can use the same or different additional sources of options and controls. This source is scanned after and can override options and controls read from the common SJJSPARM library members.

## JOB/SCAN Panel Fields

This section discusses the additional source of options and controls used by JCL Validation run from the JOB/SCAN panels. Other sections, for instance, compare the Edit macro and batch modes of operation with the panel mode, and discuss means of ensuring consistent JCL Validation results.

The JOB/SCAN - JCL Validation panel and the JOB/SCAN Processing Options panel fields establish values for many of the Report Options and Run Options. These values override corresponding values contained in the SJJSPARM options members.

For the correspondence of the panel fields and SJJSPARM options, see [Relating JCL Validation Panel Fields to SJJSPARM Options](#) and [Relating Processing Options Panel Fields to SJJSPARM Options](#).

The JOB/SCAN - JCL Validation panel also contains a First PROCLIB field that can specify a dataset to be allocated as file DD JJJPC for JOB/SCAN to search before any other PROC libraries when validating JCL.

## Verifying Panel Controls

To verify the SJJSPARM library options members and the options and controls set by the JOB/SCAN - JCL Validation panel and JOB/SCAN Processing Options panel, run the JCL Validation job from panel-assisted batch, and then SJJSPARM library browse the dataset specified by the DSN parameter of the File Allocations control statement for ddname JLINTRDR in the appropriate SJJSPARM library options member (usually JSOPT02). The SJJSPARM options members used are named in parameters of the EXEC statement, and the options and controls generated from the JOB/SCAN - JCL Validation panel and JOB/SCAN Processing Options panel are shown as SYSIN data, except for First Proclib entries which appear as //JJPC DD statements.

In the below figure, the values for EXEC statement parameters OPT01, OPT02, and OPT0N, are names of the SJJSPARM library options members used for the JCL Validation job. In this example, the member names are the names of the default members, JSOPT01, JSOPT02, and JSOPT03. The DD statement for STEP01.JJJPC shows the effect of entering 'DSSI.COM2TEST.PROCLIB' in the First Proclib field of the JOB/SCAN - JCL Validation panel. Settings for other fields in the JOB/SCAN - JCL Validation panel and the JOB/SCAN Processing Options panel produce the control statements in the SYSIN data.

### JOB/SCAN - JCL Validation Panel before Member Selection

```

----- JOB/SCAN - JCL VALIDATION ----- PROCESS COMPLETE
COMMAND ===> F

      F - Foreground execution mode          P - Processing Options
      B - Batch execution mode              E - Edit JCL member
      S - Submit After Validation           BR - Browse JCL member
Input Library  ===> 'ASGCOM2.BOOK.JOBS'
Input Type    ===> J                      (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ===> PDS                    (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ===>
Exclude Members ===>
First Proclib ===>
Proclib Type  ===>                      (PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
Report Heading ===>
Display       ===> Y (Yes/No/ErrorsOnly)  Browse/View ===> B (B/V)
Hardcopy      ===> Y (Yes/No/ErrorsOnly/SpaceOnly)
Options in Effect Listing ===> N (Y/N)      SJL PDSOUT ===> N (Y/N)

Selection Exit ===>                      Parm    ===>
Standards Name ===>                      Parm    ===>

```

**Sample JLINTRDR File**

```

BROWSE      ASGCOM2.WORK.INTRDR                      Line 00000000 Col 001 080
Command ==>                                         Scroll ==> CSR
***** Top of Data *****
//ASGCOM2A JOB (DSSICS,132900,ASG,37), 'COM2TESTER',
//          NOTIFY=ASGCOM2,MSGCLASS=X
//*
/*ROUTE PRINT U609
//JOBPROC DD DSN=DSSI.#COM2.PROCLIB,DISP=SHR
//STEP01 EXEC JS6PROC,OPT01=JSOPT01,OPT02=JSOPT02,PRM=S,
//          OPT0N=JSOPT03
//STEP01.JLJPC DD DISP=SHR,
//          DSN=DSSI.COM2TEST.PROCLIB
//STEP01.SYSIN DD *
REPORT OPTIONS
      OPTIONS IN EFFECT=PRINT
      STRUCTURED JCL DISPLAY=ON
      STRUCTURED JCL LISTING=PRINT,ERRORTEXT=EXTENDED
RUN OPTIONS
      ACCESS CONTROL CARDS =YES,LIMIT=0
      CATALOG ACCESS =YES,VTOC,PDSMEM,PGM
      JCL CHANGE PARAMETERS=STRCHG80
      PREFIX =ASGCOM2
      GROUP ID =C#132900
      USER ID =ASGCOM2
INPUT SELECTION
      SELECT=GOJLJPA
      DSNAME=ASGCOM2.TEST.CNTL
      SOURCE TYPE=PDS

```

## Specifying a First Proclib for JCL Validation

When validating JCL, it might be necessary to specify a PROC library to be scanned before any other PROC libraries (for example, to test a development PROC that is to replace an existing PROC in a production job). The JOB/SCAN - JCL Validation panel (figure - [JOB/SCAN - JCL Validation Panel](#)) provides a First Proclib field and an associated Proclib Type field for this purpose.

For example, consider the production job GOFORIT (shown in figure - [DSSI.PROD.CNTL\(GOFORIT\) -- Job Stream to Be Scanned](#)), which resides in library PROD.CNTL. A new PROC062 member, for performing the functions of STEP01, needs to be validated in its target environment before being promoted to the production system. That is, the new PROC062 JCL must be validated with the JCL of any unchanged production members and installation defaults.

**DSSI.PROD.CNTL(GOFORIT) -- Job Stream to Be Scanned**

```

//GOFORIT JOB (ACCTG), 'DO IT TO IT',CLASS=A
//JCLLIB DD DSN=PROD.PROCLIB,DISP=SHR
//STEP01 EXEC PROC062
//STEP02 EXEC PROC067

```

This example further assumes that the development version of the PROC062 PROC is the only member in library TEST.PROCLIB. The production versions of all PROCs required by the GOFORIT job, including the old version of PROC062,

are members in library PROD.PROCLIB. Note the comments on the EXEC statements of PROC062 members, shown in the below figures, indicating PROC version numbers.

#### DSSI.PROD.PROCLIB(PROC062) -- PROC062 in the Production PROC Library

```
//PROC062  PROC
//PSTEP1  EXEC  PGM=GRPT023      PROC062 - VERSION 003Z
//DD1     DD   DSN=DEPT03.GOVARS,DISP=SHR
//DD2     DD   DSN=DEPT03.GORPT01,DISP=(MOD,PASS)
```

#### DSSI.TEST.PROCLIB(PROC062) -- PROC062 in the Test PROC Library

```
//PROC062  PROC
//PSTEP1  EXEC  PGM=GRPT024      PROC062 - VERSION 004A
//DD1     DD   DSN=DEPT03.GOVARS,DISP=SHR
//DD15    DD   DSN=DEPT03.GODATA03,DISP=SHR
//DD2     DD   DSN=DEPT03.GORPT01,DISP=(MOD,PASS)
```

#### To validate the entire GOFORIT job, including the new PROC062 member

1. On the JOB/SCAN - JCL Validation panel, in the First Proclib field type the name of the library that must be scanned first because it contains the new PROC062 member (for example, 'DSSI.TEST.PROCLIB').
2. In the Proclib Type field, type PDS for partitioned dataset.
3. Fill in the remaining fields as appropriate. The JOB/SCAN - JCL Validation panel now looks similar to the below figure.
4. Press Enter to execute JOB/SCAN JCL Validation.

#### JOB/SCAN - JCL Validation Panel

```
----- JOB/SCAN - JCL VALIDATION -----
COMMAND ==> F

  F - Foreground execution mode          P - Processing Options
  B - Batch execution mode              E - Edit JCL member
  S - Submit After Validation          BR - Browse JCL member
Input Library ==> 'DSSI.PROD.CNTL'
Input Type   ==> J                      (J - JOB JCL, P - PROC, or L - Selection List)
Library Type ==> PDS                    (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> GOFORIT
Exclude Members ==>
First Proclib ==> 'DSSI.TEST.PROCLIB'
Proclib Type ==> PDS                    (PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
Report Heading ==>
Display       ==> Y (Yes/No/ErrorsOnly)  Browse/View ==> B (B/V)
Hardcopy      ==> N (Yes/No/ErrorsOnly/SpaceOnly)
Options in Effect Listing ==> N (Y/N)      SJL PDSOUT ==> N (Y/N)

Selection Exit ==>                      Parm ==>
Standards Name ==>                      Parm ==>
```

5. JOB/SCAN displays the Structured JCL Listing. The comment on the EXEC statement of PROC062 confirms that the development version of PROC062 was used for the JCL Validation.

## Structured JCL Listing Panel

```

JOB/SCAN          STRUCTURED JCL LISTING      6.1.2B          11:02:53
-----
//GOFORIT JOB (ACCTG), 'DO IT TO IT', CLASS=A
//PROCLIB DD DSN=DSSI.PROD.PROCLIB, DISP=SHR
//STEP01 EXEC PROC062
  P1//PROC062 PROC
  P1//PSTEP1 EXEC PGM=GRPT04          PROC062 - VERSION 004A
  P1//DD1 DD DSN=DSSI.DEPT03.GOVARS, DISP=SHR
  P1//DD15 DD DSN=DSSI.DEPT03.GODATA03, DISP=SHR
  P1//DD2 DD DSN=DSSI.DEPT03.GORPT01, DISP= (MOD, PASS)
//STEP02 EXEC PROC067
  P1//PROC067 PROC
  P1//PSTEP2 EXEC PGM=GSUM014          PROC067 - VERSION 001M
  P1//DD1 DD DSN=DSSI.DEPT03.GORPT01, DISP= (MOD, PASS)
*** ERROR SUMMARY AND COUNTS ***
  0 ADVISORY LEVEL
  0 WARNING LEVEL
  0 ERROR LEVEL
  0 TOTAL ISSUED          0 SUPPRESSED.
  THE HIGHEST SEVERITY CODE ENCOUNTERED WAS 00.
...

```

## Searching a PROC

This section describes how JOB/SCAN JCL Validation simulates the MVS method of searching for PROCs.

### MVS PROC Search Order

To expand a cataloged PROC encountered in the job stream, MVS scans PROC libraries until it locates a PROC of the correct name (or has exhausted the PROC libraries available). MVS first scans datasets explicitly and dynamically allocated to PROC libraries by the job stream JCL (JOBPARM statement, JCLLIB statement, etc.). If the PROC is not found, then MVS searches the installation default PROC library, typically defined by the PROC00 DD statement in SYS1.PROCLIB member JES2.

### JOB/SCAN PROC Libraries

JOB/SCAN can allocate PROC datasets to three ddnames:

DDname	Description
JLJPA	Datasets specified in File Allocations control statements in a PARMLIB options member (normally the same datasets as in the MVS installation default PROC library).
JLJPB	Datasets explicitly and dynamically allocated to PROC libraries by the input job stream JCL. JOB/SCAN dynamically allocates them to ddname JLJPB.
JLJPC	When running JCL Validation from the ASG-JOB/SCAN - JCL Validation panel, if the First Proclib field and associated Proclib Type field of the JOB/SCAN - JCL Validation panel specify a PROC library to be searched before any other PROC libraries, JOB/SCAN allocates that dataset to ddname JLJPC. (See <a href="#">Specifying a First Proclib for JCL Validation</a> for more information about using a First Proclib.)



### JOB/SCAN PROC Search Order

JOB/SCAN JCL Validation normally imitates the MVS PROC search method by first searching for a PROC in datasets allocated to JLJPB (datasets explicitly and dynamically allocated to PROC libraries by the input job stream JCL). If the PROC is not found, JOB/SCAN searches the datasets allocated to ddname JLJPA.

If JOB/SCAN has allocated a First PROCLIB to ddname JLJPC, then JOB/SCAN JCL Validation searches datasets allocated to JLJPC for PROCs before performing the normal search of JLJPB and JLJPA.

In summary, JOB/SCAN searches for a required PROC in files allocated to ddnames JLJPC, JLJPB, and JLJPA in this order:

- JLJPC from First PROCLIB specified on the JOB/SCAN - JCL Validation panel.
- JLJPB from JOBPARM statement, JCLLIB statement, etc.
- JLJPA from File Allocations statements contained in a PARMLIB options member.

## Validating JCL that Includes CA-7 #J Statements

The CA-7 #J statements must be preprocessed before running JOB/SCAN JCL Validation. ASG provides two sample programs in the SJJSSAMP library to preprocess JCL containing #J statements for the CA-7 Scheduler. CA7J1 can process a single schedule ID, while CA7J2 can process from one to five schedule IDs. Except for the number of schedule IDs handled, the programs perform the same function. See [Selection Exit Interface](#) for detailed information about modifying the sample programs for your requirements.

See [Handling #J Cards with the CA-7 Scheduler Interface](#) for detailed information on how to invoke and pass parameters to the sample CA7J1 and CA7J2 programs provided by ASG.

### What the Preprocessor Does

When the preprocessor program (either CA7J1 or CA7J2) encounters a #JI or #JO statement, it compares the schedule ID and beginning and ending dates and times of that statement with values specified in the input parameter. Based on this comparison, the program either includes or omits the following statements, up to the next #JI, #JO, or #JEND statement. The program adds the prefix '//\*' to each omitted statement and each #JI, #JO, and #JEND statement, so that they will be treated as comments and not be flagged as errors by JCL Validation.

### Invoking the Preprocessor

#### *To invoke the preprocessor (either CA7J1 or CA7J2)*

1. On the JOB/SCAN - JCL Validation panel, type the program name in the Selection Exit field.
2. Type the preprocessor parameter in the corresponding Parm field. This example includes JCL between schedule IDs 10294332 and 12694357.

**JOB/SCAN - JCL Validation Panel Specifying CA-7 #J Card Preprocessing**

```

----- JOB/SCAN - JCL VALIDATION -----
COMMAND ===> F

      F - Foreground execution mode          P - Processing Options
      B - Batch execution mode              E - Edit JCL member
      S - Submit After Validation           BR - Browse JCL member
Input Library  ===> BOOK.TESTLIB
Input Type    ===> J                      (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ===> PDS                    (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ===> CA7TEST
Exclude Members ===>
First Proclib ===>
Proclib Type  ===>                      (PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
Report Heading ===> JCL VALIDATION PANEL
Display        ===> Y (Yes/No/ErrorsOnly) Browse/View ===> B (B/V)
Hardcopy       ===> N (Yes/No/ErrorsOnly/SpaceOnly)
Options in Effect Listing ===> N (Y/N)      SJL PDSOUT ===> N (Y/N)

Selection Exit ===> CA7J2          Parm  ===> 10294332,12694357

Standards Name ===>              Parm  ===>

```

**To invoke the preprocessor from a batch job**

- Include Input Selection control statements in the JLRCL ddname similar to those shown in the below figure.

**Input Selection Control Statements Specifying CA-7 #J Card Preprocessing**

```

INPUT SELECTION
SELECTION EXIT=CA7J2,'102943320830943321730,126943570830943571730'

```

**Specifying the Parameter**

Specify the parameter in the form: *iiyydddhmmmyyddhhmm*

where:

- *iii* is the schedule ID
- *yy* is the year
- *ddd* is the Julian day of the year
- *hh* is the hour (from 24-hour clock)
- *mm* is the minute
- The first date and time in the parameter is the beginning date and time; the second is the ending date and time

For the selection to be in effect for one day only, omit the ending date from the parameter. The parameter shown in figure - [JOB/SCAN - JCL Validation Panel Specifying CA-7 #J Card Preprocessing](#) (Parm field entry) contains two schedule IDs with beginning dates only. The parameter shown in figure - [Input Selection Control Statements Specifying CA-7 #J Card Preprocessing](#) contains two schedule IDs with beginning dates and times, and ending dates and times.

The values in the input parameter are positional. If the #JI and #JO cards do not contain date or time fields, leave the date and time fields blank in the parameter. If time but not date is used, type zeros in the date field.

# Reports

This topic describes the reporting facilities that are available in JOB/SCAN.

- [Structured JCL Listing \(SJL\)](#)
- [Detail Information Section of the SJL](#)
- [Error Summary Section of the SJL](#)
- [Viewing Options in Effect](#)
- [Suppressing JOB/SCAN Error Messages](#)
- [Displaying Extended Error Messages](#)
- [JMSG: Extended Error Messages for the JEM Edit Macro](#)

## Structured JCL Listing (SJL)

The SJL presents the JCL in an easy-to-read and understand format showing all statements in a job stream in run-time sequence. All PROCs are expanded, symbolics and overrides are resolved, and optionally, control card statements included. Error messages generated by JOB/SCAN appear immediately after the statements containing the errors and in the Error Summary at the end of the report.

The format and content of the Structured JCL Listing are determined by Run Options and Report Options. The Product Administrator controls these options. However, some options can be altered in ISPF panel operation by using the Processing Options and JCL Validation panels. See [SJJSPARM Library Options Members](#) for more information.

### Generating a Report

There are two ways to generate the SJL. (See [JMSG: Extended Error Messages for the JEM Edit Macro](#) for more about the error reporting for Edit macros.) The first is using the fields and parameters on the Validation and the Processing Options panels. In this method, generally used by the Validation panel and the Edit macros, JOB/SCAN writes files to the held output. In the second method, a batch run, JOB/SCAN uses the members in the SJJSPARM library to write the SJL.

### Contents of the Structured JCL Listing

This table lists the major sections of the Structured JCL Listing.


Report Section	Description
JCL Detail Information	Complete JCL with expanded PROCs, resolved symbolics and overrides, control cards, and error messages.
Statistical Summary	Total number of PROCs, programs, DD statements, records, and input/output tape DD statements.
Space Analysis	Composite list of all new space requirements for the validated job stream. The total allocation requirements are broken down by UNIT and VOLSER and

Report Section	Description
----------------	-------------

	indicate whether a file is permanent or temporary.
Error Summary	List of errors in order of occurrence.

See [Detail Information Section of the SJL](#) and [Error Summary Section of the SJL](#) for full descriptions of the JCL Detail Information, error messages, and the Error Summary.

If the JOB/SCAN Standards facility is used, site-specific error messages can be included in the Structured JCL Listing. For more information about the JOB/SCAN Standards Facility, see [Standards Information](#).

 • When JOB/SCAN validates a PROC library, the PROC name appears on the report in place of the job name.

• On printed reports, the Statistical Summary appears after the Error Summary. This section does not appear if the report is routed to a terminal.

### Sample SJL Output

Below is the sample input JCL.

#### Input JCL

```
//PJGL0070 JOB (A,P,GL),' JONES' CLASS=P,MSGCLASS=T,
//          MSGLEVEL=(1,1)
//          JCLLIB ORDER=ASGCOM2.#JKJ.CNTL
//JS010    EXEC PPGL0070,                               EDIT AND UPDATE
//          TAPEVOL=111111,
//          TYPE=WEEKLY
//PS010.SYSPRINT DD SYSOUT=Z
//PS030.DATECARD DD *
070589     JULY 7, 1989                                1 1 3 NNN
//PS050.DATECARD DD *                                  DATE CARD OVERRID
070589     TYPE 5 CONTROL
ACCDATE=070289
AUDITDATE=070189
```

Below figures illustrate some of the SJL information that JOB/SCAN produces for the sample input JCL.

#### Structured JCL Listing: JCL Analysis

1	STRUCTURED JCL LISTING	ASG TECHNOLOGIES GROUP, INC.	PAGE	1
-----				
	DSNAME - ASGUSR.JCL	MEMBER - PJGL0070	VOLUME - DSSI05	
	JOB - PJGL0070	MSGCLASS -	AS OF: 2016/09/13	
		CLASS -	08:08:48	
	PROGRAMMER NAME -	ACCTG - ('GENERATED BY ASG-JOB/SCAN')		
-----				
	JOB JCL	PROCLIB	SORT/UTILITY CARDS	INDEX NUMBER
-----				
	//PJGL0070	JOB	('GENERATED BY ASG-JOB/SCAN')	1
0	//	EXEC	PJGL0070 ('GENERATED BY ASG-JOB/SCAN')	1 1
	P1//PJGL0070	JOB	(A,P,GL),AJONESCLASS=P,MSGCLASS=T,	1 1
	P1//	MSGLEVEL=(1,1)		1 1
	***ERROR	- DSS1125E	- JOB STATEMENT INVALID IN PROCEDURE	1 1
	P1//	JCLLIB	ORDER=ASGCOM2.#JKJ.CNTLDISP=OLD	1 1
	***ERROR	- DSS8540E	- "JCLLIB" MUST BE CODED BEFORE THE FIRST "EXEC"	1 1
	***ERROR	- DSS8530E	- INVALID SYNTAX IN "JCLLIB" STATEMENT	1 1
0	P1//JS010	EXEC	PPGL0070,                               EDIT AND UPDATE	1 2

## Structured JCL Listing: Statistical Summary

```

0JOB (PJGL0070) CONSISTS OF      2 PROCS,      0 PROGRAMS,      0 DD STATEMENTS, AND      14 TOTAL RECORDS.
                       THERE ARE      0 INPUT AND      0 OUTPUT      TAPE DD STATEMENTS.

```

## Structured JCL Listing: Error Summary

```

**** ERROR SUMMARY AND COUNTS ***
0  0  ADVISORY LEVEL
0  0  WARNING LEVEL
10 0  ERROR LEVEL
10 0  TOTAL ISSUED      0 SUPPRESSED.
0  THE HIGHEST SEVERITY CODE ENCOUNTERED WAS      08.
OPAGE  SV MSG.NO.  ERROR MESSAGE                                JOBNAME  PROCNAME  STEPNAME  DD-NAME  INDEX NUMBER
-----
0  1  8 DSS1125E - JOB STATEMENT INVALID IN PROCEDURE          PJGL0070 PJGL0070          1  1
1  8 DSS8540E - "JCLLIB" MUST BE CODED BEFORE THE FIRST "EXEC" PJGL0070 PJGL0070          1  1
1  8 DSS8530E - INVALID SYNTAX IN "JCLLIB" STATEMENT          PJGL0070 PJGL0070          1  1
1  8 DSS6090E - "*R14 " NOT DELIMITED BY EQUAL SIGN          PJGL0070 PPGL0070          1  2
1  8 DSS2020E - PROCEDURE "PPGL0070" NOT FOUND                PJGL0070 PJGL0070 JS010          1  2
1  8 DSS2010E - OVERRIDDEN STEP "PS010 " NOT FOUND IN PROCEDURE PJGL0070 PPGL0070          1  2
1  8 DSS2010E - OVERRIDDEN STEP "PS030 " NOT FOUND IN PROCEDURE PJGL0070 PPGL0070          1  2
1  8 DSS2010E - OVERRIDDEN STEP "PS030 " NOT FOUND IN PROCEDURE PJGL0070 PPGL0070          1  2

```

## Structured JCL Listing: Run Stats

```

-----
|JOB - RUNSTATS                                MSGCLASS -                                AS OF: 2007/02/13 |
|                                           CLASS -                                08:08:48 |
|PROGRAMMER NAME -                            ACCTG -                                |
-----
**** INPUT STATISTICS ***
0  TOTAL JCL RECORDS      -      13
   FROM CARD INPUT        -      2
   FROM PROCLIB           -     11
   FROM PARM LIBS         -      0
JOBS PROCESSED            -      1
PROCEDURES PROCESSED     -      2
STEPS PROCESSED           -      0
DD STATEMENTS PROCESSED  -      0
AVERAGE PROCS/JOB       -      2.0
AVERAGE STEPS/PROC      -      .0
AVERAGE STEPS/JOB       -      .0
AVERAGE DDS/STEP        -      .0
AVERAGE DDS/PROC        -      .0
USAGE REPORTING RECORD WRITTEN
0  XPJ60K - END OF REPORT      ASG-JOB/SCAN      RELEASE JS7.0.0

```

## Detail Information Section of the SJL

The Detail Information is the main body of the SJL and includes all JCL statements with symbolics and overrides resolved. The Detail Information also expands any cataloged procedures and lists control cards and normal, extended, and site-specific error messages.

### Index-Numbering JCL Statements

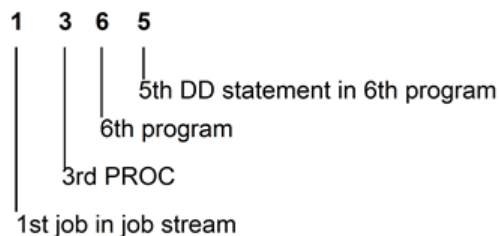
JOB/SCAN identifies all statements within the JCL being validated using index numbers. These numbers, which appear on the 132-character-wide version of the SJL, identify the statements' positions in the overall JCL structure. Use these numbers as cross-references between the Error Summary and the Detail Information sections.

An index number consists of these parts:

- Job statement sequence number within the job stream.
- PROC sequence number (increments for each PROC, including nested PROCs).

- Program sequence number (increments for each program, including programs from INCLUDE statements).
- DD statement sequence number (within the program indicated by the program sequence number).

For example:



### Handling Symbolics

JOB/SCAN validates symbolics and applies them to each appropriate line in the procedure. Lines that require symbolic substitution are listed first in raw form (lines begin with //), followed immediately by a line showing the replacement (lines begin with --). Because JOB/SCAN applies symbolics in the same way the operating system does, JCL substitution errors are detected during validation.

### Handling Overrides and Add-ons

JOB/SCAN lists DD statement overrides and add-ons as it encounters them after the EXEC statement. Overrides and add-ons (see figure - [Example SJL showing disposition of DD overrides and add-ons](#)) are aligned with the EXEC statement and are preceded by \$\$ instead of the usual //. When the overridden or added-on ddname is encountered in the referenced step, JOB/SCAN applies the override or add-on. JOB/SCAN prints the overriding or added-on statement, prefixing it with \$/, immediately after the statement to which the override or add-on applies.

### Including Control Card Information in Reports

#### ***To include control card information on JOB/SCAN reports***

1. Turn on the Access Control Cards Run Option in JSOPT02.
2. Turn on catalog and VTOC access using the Catalog Access Control Run Option.
3. If a JCL statement references a member in a PDS library, specify either the appropriate ddnames in Control Table AB or the dataset names in Control Table AQ.
4. If preproduction control card libraries should be searched for the member names instead of the library specified in the JCL, code a JLJPF allocation in the File Allocations section of the PARMLIB. This ddname should point to the preproduction library where control card members reside.

**Example SJL showing disposition of DD overrides and add-ons**

```

***** Top of Data *****
JOB/SCAN          STRUCTURED JCL LISTING  6.2.2A   12/22/2004  11:12:30
-----
//MYJOB          JOB  (JRR) , 'UNLOAD' ,MSGCLASS=X,          000100
//              NOTIFY=ASGUSR,MSGLEVEL=(1,1)              000110
//PROC1          PROC                                     000200
//S1             EXEC PGM=IEFBR14                          000300
//DD1           DD   DISP=SHR,DSN=ASGUSR.JCL.TEST         000310
//             PEND                                       000400
//TEST1          EXEC PROC1                               000500
$$S1.DD1        DD   DISP=SHR,DSN=ASGUSR.JCL.PROD        000600
$$S1.DD2        DD   SYSOUT=*                            000700
  P1//PROC1      PROC                                     00
  P1//S1         EXEC PGM=IEFBR14                         00
  P1//DD1        DD   DISP=SHR,DSN=ASGUSR.JCL.TEST       00
  P1$/S1.DD1     DD   DISP=SHR,DSN=ASGUSR.JCL.PROD       00
  P1//          PEND                                       00
  P1$/S1.DD2    DD   SYSOUT=*                            00
...

```

**Handling Control Cards**

Optionally, JOB/SCAN lists control cards after an input DD statement. Parameters defined by the Product Administrator determine whether and how many, control cards print in the SJL.

**Example Input JCL for JOB/SCAN JCL Validation**

```

//PPJL0070 PROC  APPLIC=GLPROD,
//              DVOL=VOLWEEK2,
//              GEN3='+1',
//              GEN4='+1',
//              GEN5='+1',
//              INDX=DSSI,
//              KTDSN=KEYTAPE,
//              KTUNIT=CART,
//              SPROLL=500,
//              SPTRN=28,
//              TAPEVOL=NEEDED,
//              TYPE=D
//PS010         EXEC  PGM=IEBGENER                        COPY THE KEYTAPE TO DISK
//SYSPRINT      DD   SYSOUT=A
//SYSUT1        DD   DSN=&KTDSN,
//              DISP=OLD,
//              UNIT=&KTUNIT,
//              LABEL=(,NL),
//              VOL=SER=&TAPEVOL,
//SYSUT2        DD   DSN=&TEMP,
//              DISP=(,PASS)

```

**Format of the SJL**

As shown in below figures, job-level JCL statements start with the job statement at the left margin; called procedures are indented to the right. If control cards are printed, JOB/SCAN indents them further to follow their input DD statement.

Example Structured JCL Listing (Part 1 of 2)

Job stream data	Override statement	PROC-level statement			
STRUCTURED JCL LISTING 1 PAGE					
DSNAME - DSSCOM2.HJKJ.ZHTL MEMBER - PJGLE070 VOLUME - DSS001					
JOB - PJGLE070		MSGCLASS - 1	AS OF: 09/15/94		
PROGRAMMER NAME - JONES		CLASS - 3	14:20:05		
		ACCTG - (A,F,GL)			
JOB JCL	PROCLID	SORT/UTILITY CARDS			INDEX NUMBER
//PJGLE070 JOB (A,P,EL), 'LINES', CLASS=M, MSGCLASS=M,					
		MSGLEVEL=(1,1)			1
		JCLLIB ORDER=DSSCOM2.HJKJ.CNTL			1
		//SYSAF EXFC PDAI RA78		FD1 AND HFDATF	1
		// TAPEVCL=111111,			1
		// TYPE=MEEKLY			1
		\$SPS013 SYSPR:HT DD SYSOUT=Z			1
		\$SPS033 DATECARD DD *			1
		070509 JUL 7, 1969 1 1 3 MIN			1
		\$SPS053 DATECARD DD *		DATE CARD OVERRIDE	1
		RA78RQ		TYPE=CONTROL	1
		ACCDATE=070239			1
		AUDITDATE=073100			1
		F1//PJGLE070 PROC APPLIC=GLPRD0,			
		F1// DVOL=VOLWEEK2,			
		F1// GE13='+1'			
		F1// GE14='+1'			
		F1// GE15='+1'			
		F1// INDX=DSSI			
		F1// KTDISN=KEYTAPE,			
		F1// KTDJN=TCART,			
		F1// SP201=53R,			
		F1// SP202=20,			
		F1// TAPEVCL=NEEDED,			
		F1// TYPE=Q			
		P1//PG31 E CNCC PG4=ICD3CHER		COPY THE KEYTAPE TO DISK	1 1 1
		P1//SYSPRINT DD SYSOUT=A			1 1 1 1
		P1//PG31 E,SYSPRINT DD SYSOUT=Z			1 1 1 1
		P1//SYSUT1 DD DSN=KTDG4		(FJG_TR01) FROM DATA PREP	1 1 1 2
		P1 SYSUT1 DD DSN=CYTAPE,		(PJGLTR01) FROM DATA PREP	
		..			
		P1// D:SP=31R			
		***ERROR - DSS4050E - ENTR SET NOT FOUND IN CATALOG			1 1 2 1
		*REVISORY - DSS8986E - CSN = DSS1.SPRD.LJAD			1 1 2 1
		*REVISORY - DSS2857A - I TRAEV NOT SEARCHED FOR PROGRAM NAME			1 1 2 1
		P1//SORTLIB DD DSN=SYS1.SORTLIB,		-FD SORT RESIDENCE	1 1 2 1
		P1// D:SP=31R			1 1 2 2
		P1//SORTPRT DD SYSOUT=(J,9),		(PJGLR05) SORT STATISTICS REPORT	1 1 2 2
		P1// LUP=ES=2			1 1 2 3
		P1//SYSPRINT DD SYSOUT=A			1 1 2 3
		P1//SYSIN DD DSN=8INDX..&APPLIC..CNTL(G2010015),		SORT CTL	1 1 2 4
		P1--SYSIN DD DSN=DSS01.GLPRD0.CNTL(G2010315),		SORT CTL	1 1 2 5

**Error message** (points to error lines)

**Override applied to proper statement** (points to F1//PJGLE070 PROC)

**Statement with symbolics resolved** (points to P1//SYSUT1 DD DSN=KTDG4)

Example Structured JCL Listing (Part 2 of 2)





- Error number DSS4050E has a severity level of 8.
- Error number DSS8900A has a severity level of 0.

Using Control Table AW, the Product Administrator can alter the severity level associated with different messages or can suppress error messages altogether. See [Suppressing JCL Validation Error Messages](#) for further information.

**Example Structured JCL Listing Error Summary**

```

0*** ERROR SUMMARY AND COUNTS ***
0 6 ADVISORY LEVEL
0 2 WARNING LEVEL
0 7 ERROR LEVEL
0 15 TOTAL ISSUED      0 SUPPRESSED.
0 THE HIGHEST SEVERITY CODE ENCOUNTERED WAS 00.
0PAGE SV MSG. NO.  ERROR MESSAGE                                JOBNAME  PROCNAME  STEPNAME DD-NAME  INDEX NUMBER
-----
0 1 4 DSS4975W - VOLUME "111111" NOT AVAILABLE                    PJGL0070 PPGL0070 PS010 SYSUT1    1 1 1 2
0 1 8 DSS4940E - UNABLE TO IDENTIFY UNIT NAME "CART "            PJGL0070 PPGL0070 PS010 SYSUT1    1 1 1 2
0 2 8 DSS4050E - DATA SET NOT FOUND IN CATALOG                  PJGL0070 PPGL0070 PS020 STEPLIB   1 1 2 1
0 2 0 DSS8900A - DSN = "DSS1.$PRD,LOAD "                          PJGL0070 PPGL0070 PS020 STEPLIB   1 1 2 1
0 2 0 DSS2052A - LIBRARY NOT SEARCHED FOR PROGRAM NAME          PJGL0070 PPGL0070 PS020 STEPLIB   1 1 2 1
0 2 8 DSS5005E - GDG INVALID AS REFERENCED IN VOL=REF          PJGL0070 PPGL0070 PS020 SORTOUT    1 1 2 7
0 2 0 DSS9000A - DSN = "DSS1.GLPROD.VOLMEK2 "                   PJGL0070 PPGL0070 PS020 SORTOUT    1 1 2 7
0 2 8 DSS4420E - GENERATION DATA GROUP DOES NOT EXIST         PJGL0070 PPGL0070 PS020 SORTOUT    1 1 2 7
0 2 0 DSS8900A - DSN = "DSS1.MDGL0027(+1) "                     PJGL0070 PPGL0070 PS020 SORTOUT    1 1 2 7
0 3 4 DSS4260W - BLKSIZE IS NOT A MULTIPLE OF LRECL IN RECFM=FB PJGL0070 PPGL0070 PS030 MASTOUT   1 1 3 5
0 3 8 DSS4050E - DATA SET NOT FOUND IN CATALOG                  PJGL0070 PPGL0070 PS030 TRANS     1 1 3 6
0 3 0 DSS8900A - DSN = "DSS1.GLPROD.MDGL0027(+1) "              PJGL0070 PPGL0070 PS030 TRANS     1 1 3 6
0 3 8 DSS5005E - GDG INVALID AS REFERENCED IN VOL=REF          PJGL0070 PPGL0070 PS040 ROLLTOUT  1 1 4 5
0 3 0 DSS9000A - DSN = "DSS1.GLPROD.VOLMEK2 "                   PJGL0070 PPGL0070 PS040 ROLLTOUT  1 1 4 5
0 4 8 DSS2050E - PROGRAM "PXGLSMRY" NOT FOUND                   PJGL0070 PPGL0070 PS050 ROLLTOUT  1 1 5
0XPJ60K - END OF REPORT JOB/SCAN 6.0.4M (C) DIVERSIFIED SOFTWARE SYSTEMS
    
```

Detected Advisories, Warnings, and Errors

Errors suppressed

Highest severity level

Summary of JCL error messages issued

## Viewing Options in Effect

This section describes how to produce an online display or a report showing the options in effect for JOB/SCAN.

### Viewing Options in Effect on Line

The Options in Effect Report is available only in hardcopy form. You can use these commands to view the options and tables:

Command	Description
SHOW OPTIONS	An ISPF command that can be used on any JOB/SCAN panel to view the Run Options, Report Options and Control Tables in effect. (JOB/SCAN help panels are not a valid environment for this command.)
SHOW RUN	An ISPF command that can be used on any JOB/SCAN panels to view only the Run Options. (JOB/SCAN help panels are not a valid environment for this command.)
SHOW REPORT	An ISPF command that can be used on any JOB/SCAN panels to view only the Report Options. (JOB/SCAN help panels are not a valid environment for this command.)
SHOW TABLES	An ISPF command that can be used on any JOB/SCAN panel to view only the Control Tables in effect. (JOB/SCAN help panels are not a valid environment for this command.)

## Options in Effect Report

The Options in Effect Report lists all Run Option, Report Option, File Allocations, and Control Table parameters in effect during the current JOB/SCAN Validation run. This is a troubleshooting report, useful for monitoring parameter changes to SJJSPARM library members. See [Reference Information](#) for complete information about the SJJSPARM library, Run and Report Options, and Control Tables.

The Options in Effect Report flags any errors associated with coding options and displays an error message immediately after the statement causing the error. All such errors must be corrected before continuing validation.

There are two versions of the Options in Effect Report: a summary-level report, and a detail-level report. The Product Administrator determines which version is produced. Figure - [Sample Options in Effect Report](#) shows a summary-level Options in Effect Report. This table describes the two versions of the report.

Report Version	Description
Detail-level report	Prints the contents of all SJJSPARM library members read by JOB/SCAN.
Summary-level report	Includes the Run Options and Report Options actually in effect for the current run and does not print File Allocations statements, Access Control statements, or Control Tables.

## Requesting the Report from the Panels

### *To request an Options in Effect report for a JCL Validation run from the JOB/SCAN - JCL Validation panel*

- Type Y in the Options in Effect field.

### *To produce an Options in Effect Report every time the JOB/SCAN panels are invoked from ISPF*

1. Select Hardcopy Output and JOB Statement Definition from the ASG-JOB/SCAN Primary Menu and press Enter. The Hardcopy Output Characteristics panel appears.
2. Type Y in the Force OPE Hardcopy for Initialization field and press Enter.
3. Exit JOB/SCAN.

Each subsequent time that JOB/SCAN panels are invoked, JOB/SCAN produces an Options in Effect Report that includes the contents of JSOPT01 and JSOPT02.

## Requesting the Report in Batch

### *To produce the Options in Effect Report in pure batch mode*

- Specify the PRINT, DETAIL, TABLES parameters in the Options in Effect Report Option in the SYSIN stream.

Or

- Ask the Product Administrator to specify it in the appropriate SJJSPARM library member.

See [Report Option Group](#) and [Modifying SJJSPARM Members](#) for complete information about Report Options and modifying SJJSPARM library members.

## Sample Options in Effect Report

```

OPTIONS IN EFFECT                ASG TECHNOLOGIES GROUP, INC.
-----
*** SUMMARY OF OPTIONS FOR THIS RUN ***
*****
RUN OPTIONS
*****
ACCESS CONTROL CARDS             =YES, LIMIT=015
CATALOG ACCESS                   =YES, VTOC, CVOL, PGM, PDSMEM
DEFAULT UNIT NAMES               =CARTRIDGE=3480, DASD=DISK,
                                MASSTOR=SYSDAV, TAPE=TAPE
DEVICE CODES                     =DASD=, REPORT=, TAPE=, SMS=
DMS VOLUME SERIAL                =ARCIVE
STANDARDS NAME                   =
GDG PROCESSING                   =ENDOFJOB
GROUP ID                          =
HSM VOLUME SERIAL                =MIGRAT
ISPF SPECIFICATIONS              =NOAUTHFUNCTIONS, DSNFORMAT=TSO
JES2 COMMAND CHARACTER           =\$
JLJPA PROCLIB TYPE               =PDS
JLJPC PROCLIB TYPE               =PDS
JOB LIMIT                        =
JOB/SCAN PROC                    =JS6PROC
LINECNT                          =0060
MIGRATE DEVICE TYPE              =
OPERATING SYSTEM LEVEL           =BCP=H, DFP=F
OUTPUT SPECIFICATIONS             =NOBANNERPAGE, STANDARD, FOOT, NUM, HEADS, NOPDSPACK,
                                NOUTPACK, NOEURODATE
OVERRIDE INDICATOR               =\$
PREFIX                           =
PROCESSING SPECIFICATIONS         =NOCONDCODE
PROGRAM PRODUCT INTERFACE         =NOACF2, NOALTPGM, NOCA1, NOCONTROL-M, DB2, NODMS,
                                EZPROCLIB, HSM, IDCAMS, (IMS, imsid), NOPANEXEC,
                                NOTMSSLOT
REPORT HEADING                   ='JSST907   JLOPE / $STG'
SCAN JCL ONLY                     =NO
SECURITY EXIT                     =SAF2, NODSNAME
SMS                               =NOACS, NOCAS
SPAN MEMBERS                      =NO
SUBSYSTEM                        =JES2
SUPPRESS ERROR MESSAGES           =NO
UPDATED JCL FILE TYPE            =PDS
USE ERROR TABLE                  =NO
USE PROC EXCLUDE TABLE          =NO
USE PROC SELECT TABLE            =NO
...

```

## Suppressing JOB/SCAN Error Messages

JOB/SCAN provides these ways to suppress error messages:

- By severity level with the Suppress Error Messages Run Option.
- For specific error codes, or combinations of specific error code and specific program name or ddname, with Control Table AW.
- According to user defined criteria with the JOB/SCAN Standards Facility.

### Suppress Error Messages (SEM) Run Option

The Suppress Error Messages Run Option suppresses, by severity level, error messages that are normally displayed in the

Structured JCL Listing. If the Suppress Error Messages Run Option is in effect, JOB/SCAN suppresses both JOB/SCAN and Standards Facility messages. The format of the Suppress Error Messages Run Option is:

```
SUPPRESS ERROR MESSAGES =NO|A|W|E|nn
```

This table describes the severity level of messages suppressed by each Parameter value.

Value	Messages Suppressed
NO	None (Default)
A	Advisory
W	Advisory and warning
E	Advisory, warning, and error
nn	All messages with a severity level equal to or less than <i>nn</i> ( <i>nn</i> = a decimal value from 0 to 99).

When JOB/SCAN is run from the JOB/SCAN - JCL Validation panel, the Suppress Error Messages field of the JOB/SCAN Processing Options panel determines the value in effect for the Suppress Error Messages Run Option.

JOB/SCAN normally resets the value of the Suppress Error Messages field of the JOB/SCAN Processing Options panel to NO for each JOB/SCAN session, but the SEMRESET Run Option can cancel resetting the value.

The format of the SEMRESET Run Option is:

```
SEMRESET =YES|NO
```

This table describes the purpose of each SEMRESET parameter value.

Value	Purpose
YES	Reset the value of the Suppress Error Messages field of the JOB/SCAN Processing Options panel to NO for each JOB/SCAN session. (Default)
NO	Save the value of the Suppress Error Messages field of the JOB/SCAN Processing Options panel for subsequent JOB/SCAN sessions.

### Control Table AW

Control Table AW provides the means of suppressing or changing the severity level of JCL error messages for specified JOB/SCAN error numbers. A Control Table AW entry can be coded to suppress any of these error messages or change their severity level:

- All occurrences of the specified error number.
- Only those occurrences in JCL for a specified program name.
- Only those occurrences for a specified ddname.
- Only those occurrences for a specified ddname in JCL for a specified program name.

**Related Run Option: Use Error Table (UET)**

The Use Error Table Run Option must be set to `YES` for JOB/SCAN to use the entries in Control Table AW. (The default if not coded is `NO`.)

When running JOB/SCAN from the JOB/SCAN - JCL Validation panel, the value (Y for yes, or N for no) entered from the JOB/SCAN Processing Options panel determines the setting.

When running in batch, the value set by the last occurring Use Error Table Run Option in ddname JLRCL (usually concatenated SJJSPARM library members) determines the setting.

The format of the Use Error Table Run Option is:

```
USE ERROR TABLE =YES|NO
```

**Control Table AW Format**

The format of Control Table AW entries and purpose of the fields is shown in this table.

Column	Entry
1-5	Blanks
6-7	AW (Identifies an AW entry).
8-15	JOB/SCAN error number, such as DSS5110E, to be suppressed or altered. The letter suffix indicating severity level is optional; JOB/SCAN ignores it.
16-20	Blanks
21-22	Blanks or nonnumeric characters to suppress the error message. If this entry is used to alter the severity level of the message, instead of suppressing the error message, this field will contain the numeric value (decimal) of the new severity level.
23	Blank
24-31	A program name if this entry is to suppress or alter this error message only for that program. Blank if it is to suppress or alter this error message for all programs.
32	Blank
33-40	A ddname if this entry is to suppress or alter this error message only for that ddname. Blank if it is to suppress or alter this error message for all ddnames.
24-40	(Both of the last two fields) Blanks to suppress or alter all occurrences of this error message.

**Control Table AW Search Precedence**

JOB/SCAN gives precedence to the Control Table AW entry occurring *first* when there is a conflict. For example, if two

entries in the table specify altering the severity level of all occurrences of the same error number to different values, then the entry occurring first prevails. If two entries in the table specify altering the severity level of the same error number to different values, but specify different programs or ddnames, then there is no conflict and JOB/SCAN will process both entries.

Entries that suppress error messages cannot conflict with each other. Because all error suppression entries do the same thing, but possibly in different instances, their effect can only be additive. That is, *all* of the Control Table AW error message suppression entries for a JOB/SCAN error number will take effect.

### Example Error Message Suppression Entries

The sample Control Table AW entries shown in figure - [Control Table AW Error Message Suppression Entries](#) suppress JOB/SCAN error messages as follows:

1. The first entry, for JOB/SCAN error number DSS4270, suppresses all occurrences of error number DSS4270.
2. The next two entries, for JOB/SCAN error number DSS5110, suppress DSS5110 messages only for ddnames IDCUT1 and IDCUT2 occurring in program IDCAMS.
3. The next entry, for JOB/SCAN error number DSS8110, suppresses DSS8110 messages for any ddnames, but only in program PRGALT4.
4. The next entry, for JOB/SCAN error number DSS8605, suppresses DSS8605 messages in any program, but only for ddname DD1.
5. The last three entries, for JOB/SCAN error number DSS4500, suppress all DSS4500 messages. The order of the three entries does not affect the result because there is no conflict and the effects are additive.

### Control Table AW Error Message Suppression Entries

CONTROL TABLES			
*			
	AWDSS4270	XX	
	AWDSS5110E	XX	IDCAMS IDCUT1
	AWDSS5110E	XX	IDCAMS IDCUT2
	AWDSS8110	XX	PRGALT4
	AWDSS8605	XX	DD1
	AWDSS4500	XX	PRGALT7
	AWDSS4500	XX	DD2
	AWDSS4500	XX	
*			

### Standards Facility

Except for DSS0xxx error messages, a JOB/SCAN standards program can change the severity of any JOB/SCAN messages, or suppress the messages altogether. See [Reference Information](#) and [Standards Information](#) for more information about standards program error message suppression.

## Displaying Extended Error Messages

This section describes how to display extended error messages for all modes of operation *except* for the JEM Edit macro. See [JMSG: Extended Error Messages for the JEM Edit Macro](#) for information about displaying extended error messages for JEM Edit macro results.

JOB/SCAN provides the option of including extended error messages in the Structured JCL Listing (or of using the JMSG Edit macro to display extended error messages for JEM Edit macro results) when more information than the short

messages provide is needed. ASG distributes extended error message text in two members of the SJJSPARM library, DSSMSG00 and DSSMSG90. DSSMSG00 contains the extended error message text for JOB/SCAN error messages. DSSMSG90 contains the extended error message text for user exits such as standards and security exits.

### Including Extended Error Messages in the Structured JCL Listing

Except when using the JEM Edit macro, extended error messages are controlled by the `ERRORTXT=` parameter of the Structured JCL Listing Report Option.

`ERRORTXT=SHORT` (the default) turns off the extended error messages.

`ERRORTXT=EXTENDED` causes the extended error messages to be included in the Structured JCL Listing.

The manner of invoking JOB/SCAN JCL Validation determines if the `ERRORTXT=` parameter must be set in an SJJSPARM option control statement, or in the JOB/SCAN Processing Options panel.

### JSCAN and JSUB Edit macros and Batch Processing

#### *To include extended error messages in the SJL when invoking JOB/SCAN JCL Validation in these environments*

- Code a Structured JCL Listing Report Option control statement with the `ERRORTXT=EXTENDED` parameter in an SJJSPARM library options member. For example:

```
STRUCTURED JCL LISTING =PRINT, ERRORTXT=EXTENDED
```

See [JOB/SCAN Libraries](#) for more information about SJJSPARM library options.



Changes to SJJSPARM library options will not take effect until JOB/SCAN has been initialized with the new values. After modifying SJJSPARM options, exit and re-enter JOB/SCAN to initialize JOB/SCAN with the new values.

### Foreground and Panel Assisted Batch (JOB/SCAN - JCL Validation Panel)

When submitting JCL Validation jobs, either to the foreground or batch, from the JOB/SCAN - JCL Validation panel, the value entered in the Extended Error Text field of the JOB/SCAN Processing Options panel overrides the `ERRORTXT=` parameter coded in SJJSPARM library control statements.

#### *To include extended error messages in the SJL when invoking JOB/SCAN JCL Validation in this environment*

1. Select Processing Options from either the ASG-JOB/SCAN Primary Menu panel or the JOB/SCAN - JCL Validation panel.
2. On the JOB/SCAN Processing Options panel, type `x` in the Extended Error Text field.

```
Extended Error Text      ===> X
```

3. Press End to save the option value and proceed with processing.

See [SJJSPARM Library Options Members](#) for additional information about options set from JOB/SCAN panels.

### Extended Error Messages from Standards Programs

The JOB/SCAN Standards Facility provides the capability of including extended error message text, as well as single line messages, in the Structured JCL Listing. See [Standards Facility Error Messages](#) for more information about standards



program error message suppression.

## JMSG: Extended Error Messages for the JEM Edit Macro

This section discusses how to display extended error messages in JEM Edit macro results. See [Displaying Extended Error Messages](#) for information about displaying extended error messages in other modes of operation.

The `ERRORTXT=` parameter of the Structured JCL Listing Report Option has no effect when JCL Validation is invoked by the JEM Edit macro. The JMSG Edit macro will display extended error message text for error messages displayed by JEM.

### Example of JEM Execution Results

```

EDIT ---- ASGCOM2.TEST.CNTL(TESTAW) - 01.19 ----- COLUMNS 001 072
COMMAND ===>
*****
***** TOP OF DATA *****
==MSG> *** ERROR SUMMARY AND COUNTS *** JEM 6.1.2K 05/23/95 16:40:53
==MSG> 0 ADVISORY LEVEL
==MSG> 0 WARNING LEVEL
==MSG> 5 ERROR LEVEL
==MSG> 5 TOTAL ISSUED 0 SUPPRESSED.
==MSG> THE HIGHEST SEVERITY CODE ENCOUNTERED WAS 08.
==MSG> LABEL SV MSG.NO. ERROR MESSAGE
==MSG> -----
==MSG> .JAAA 8 DSS4350E - UNIDENTIFIED VALUE FOR DISP
==MSG> .JAAA 8 DSS4500E - INVALID DATA SET NAME FORMAT IN DSN= OPERAND
==MSG> .JAAB 8 DSS4500E - INVALID DATA SET NAME FORMAT IN DSN= OPERAND
==MSG> .JAAC 8 DSS4500E - INVALID DATA SET NAME FORMAT IN DSN= OPERAND
==MSG> .JAAD 8 DSS4500E - INVALID DATA SET NAME FORMAT IN DSN= OPERAND
000001 //JS6PROC PROC DJSNAME='DSSI.$JS612K',
000002 //* OPT01=JSOPT01,
000003 // OPT02=JSOPT02,
000004 // OPT0N=JSOPT03,
000005 // PRG=J02YJS,
000006 //* DIAGOUT='SYSOUT=X,HOLD=YES',
000007 //* SOUT='X,HOLD=YES',
000008 // PRM=S
000009 //STEP01 EXEC PGM=&PRG, PARM=&PRM, REGION=6212K
000010 //STEPLIB DD DISP=SHR, DSN=&DJSNAME..SJJSLMOD
.JAAA //JLRCL DD DISP=WILD, DSN=DSSI.#COM2.DEMOLIB(TOOOOLONG)
==MSG> ***ERROR - DSS4350E - UNIDENTIFIED VALUE FOR DISP
==MSG> ***ERROR - DSS4500E - INVALID DATA SET NAME FORMAT IN DSN= OPERAND
000012 //LCRLJ DD DISP=SHR, DSN=DSSI.#COM2.SJJSPARM(&OPT02)
000013 // DD DISP=SHR, DSN=DSSI.#COM2.SJJSPARM(&OPT0N)
.JAAB // DD DISP=SHR, DSN=DSSI.#COM2.SJJSPARM(TOOOOLONG)
==MSG> ***ERROR - DSS4500E - INVALID DATA SET NAME FORMAT IN DSN= OPERAND
000015 //* DD DDNAME=SYSIN
000016 //*
...

```

### To display extended error messages for a line of JCL with one error

1. Run JOB/SCAN JCL Validation by executing the JEM Edit macro. See [Validating JCL with Edit Macros](#) for more information about executing JEM.

JEM displays JOB/SCAN messages in the ISPF Edit panel and replaces line numbers with JEM labels (.JAAA, .JAAB, etc.) on lines containing errors. (See figure - [Example of JEM Execution Results](#)).

2. Type `JMSG` on the command line.
3. Position the cursor on an error line (a line with a JEM label instead of a line number; for example, line .JAAA), and press Enter
4. The JMSG Edit macro displays JMSG Extended Message Text panel, if the selected line contained a single error. This panel displays the JOB/SCAN error number and short message text as a header, followed by the extended error message text for that error from SJJSPARM library member DSSMSG00 or DSSMSG90.
5. After examining the extended error message text, press End to return to the ISPF Edit panel.
6. To display extended error message text for other error lines, repeat from Step 2.

#### JMSG Extended Message Text Panel

```

----- - Extended Message Text - ----- ROW 1 TO 9 OF 9
COMMAND ==>                               SCROLL ==> CSR
DSS4350E      UNIDENTIFIED VALUE FOR DISP
*****
*
*  DSS4350      Unidentified value for DISP
*  The following are the values allowed in the first DISP sub-parameter:
*  "NEW", "OLD", "SHR", "SHARE", "MOD", or "," following the left
*  parenthesis. The second sub-parameter can contain "PASS", "KEEP",
*  "CATLG", "UNCATLG", or "DELETE". The third sub-parameter can contain
*  "DELETE", "KEEP", "CATLG", OR "UNCATLG".
*
*****

```

#### To display extended error messages for a line of JCL with multiply errors

1. Run JOB/SCAN JCL Validation by executing the JEM Edit macro. See [Validating JCL with Edit Macros](#) for more information about executing JEM.

JEM displays JOB/SCAN messages in the ISPF Edit panel and replaces line numbers with JEM labels (.JAAA,.JAAB, etc.) on lines containing errors. (See figure - [Example of JEM Execution Results](#)).

2. Type `JMSG` on the command line.
3. Position the cursor on an error line (a line with a JEM label instead of a line number; for example, line .JAAA), and press Enter

The JMSG Edit macro displays the JMSG Extended Message Selection panel, if the selected line contained multiple errors (for example, line .JAAA in figure - [Example of JEM Execution Results](#)).

4. Type `s` in the Sel field on this panel, to the left of each message number for which extended error message text is desired.
5. Press Enter to display the extended error messages.
6. The JMSG Extended Message Text panel will be displayed for each selected error message.
7. When all of the selected messages have been displayed, press Enter again to return to the Extended Message Selection panel.
8. Repeat from step a as many times as desired, and then press End to return to the ISPF Edit panel.
9. To display extended error message text for other error lines, repeat beginning with Step 2.

## JMSG Extended Message Selection Panel

```

----- - Extended Message Selection - ---- ROW 1 TO 2 OF 2
COMMAND ==>                                SCROLL ==> CSR

Sel  Message      Short Message Text
Number

      DSS4350E     UNIDENTIFIED VALUE FOR DISP
      DSS4500E     INVALID DATA SET NAME FORMAT IN DSN= OPERAND
*****

```



To facilitate using JMSG, set a PF key to the value JMSG. Then, instead of Step 2 and Step 3, position the cursor at the error line and press that PF key to display the JMSG panel.

To set a PF key to JMSG, enter KEYS on any ISPF command line and define the key on the PF Key Definitions and Labels panel. If JEM and JMSG are run from the ISPF Edit panel invoked by entering E on the JOB/SCAN - JCL Validation panel, the PF key value must be set by entering KEYS on the command line of a JOB/SCAN ISPF panel.

# JCL Change

This topic describes the JCL changing facility.

- [Basic Concepts](#)
- [Creating Change Command Members](#)
- [Overall Command Syntax](#)
- [JCL Change Commands Order of Application](#)
- [Selector Clause \(ON Command\)](#)
- [Verb Clause](#)
- [Setting the Scope](#)
- [Basic Change Examples](#)
- [Verifying Change Commands](#)
- [Running JCL Change](#)
- [Advanced Features](#)

## Basic Concepts

The JOB/SCAN JCL Change facility (Change) automates the larger part of JCL maintenance. Change handles:

- dataset renaming (for example, from TEST to PROD)
- complex statement adding and deleting
- parameter and subparameter adding, changing and deleting
- parameter renaming

You run Change from any JOB/SCAN mode: ISPF panels; Edit macros; batch jobs. The common elements for effecting JCL change, used by all modes, are the change commands stored in members. Use the JOB/SCAN panels as the development portal for Change command members. The Input panel contains fields for the command library and member, the input JCL library and members, and for a report heading. It also contains facilities to edit input members and to edit and verify Change command members. This facilitates the development and maintenance of Change command members.


The Output panel contains fields for specifying whether output JCL gets written and where, and whether the JOB/SCAN displays a report showing the original and changed JCL statements and records.

Use the batch jobs to process large numbers of members with tested commands and use the Edit macro JCHG to process a single member.

### ***To develop and use Change commands***

1. Validate JCL with JOB/SCAN prior to applying changes. Fix all errors.
2. Identify what statements, comments, control cards and other records need to be changed.
3. Identify the members to be changed.

4. Write commands to change them.
5. Test the commands against members and verify the results (using the simulation mode where practicable). Modify commands as needed to achieve desired changes, and retest.
6. Apply the change commands to the members.
7. Validate the changed JCL with JOB/SCAN.

 Although Change might work on JCL containing erroneous statements, the results of the changes are not necessarily valid JCL statements. JCL Change works best and most predictably when valid JCL is input.


8. Be careful not to write resultant JCL over the original input JCL. Always promote from validated datasets.

## Creating Change Command Members

To create or modify JCL Change command members, use command option M on the first ASG-JOB/SCAN - Change/Reformat JCL panel. An ISPF Edit session is used to compose the commands. After a member is created and saved in the command library, it can be modified at any time using the same procedure.

### *To create or modify change commands*

1. On the first JOB/SCAN - Change/Reformat JCL panel, type *M* on the command line.
2. In the Command Lib field, type the name of the command library.

 The Change command library must have a logical record length (LRECL) of 80.

3. In the Member Name field, type the name of the member containing the Change commands. Leave the field blank to display a directory of members.
4. Press Enter to process.

- If a Member Name was specified, an ISPF Edit panel appears.  
If the member doesn't exist, the Edit session is started on a blank member.

**Or**

- If the Member Name field was left blank, a directory of members appears. Select the desired member.

5. Using standard ISPF Edit functions, type the Change commands.
6. Press End when the command group is completed to save the member and exit.

## Overall Command Syntax

The high-level syntax for Change is:

```
selector_clause  verb_clause  delimiter
```

The collection of Change commands from selector to delimiter is a *command set*.

### Selector\_clause

The ON command governs the scope or selects the type of statements a given set of commands affect. JOB/SCAN applies, to the scope given by the ON command, the actions specified by commands up to the next END command. (If JOB/SCAN does not find an END before it reaches end-of-member (EOM), it assumes one.)

### Verb\_clause

The verb clause contains the actions that JOB/SCAN carries out on the member.

```
verb [object]  [preposition_clause]
```

With some restrictions, multiple verb clauses may appear with an ON command. Verb clauses specify whether JOB/SCAN:

- deletes or adds statements or lines
- changes a parameter, subparameter or string
- annotates the member with change information
- deletes comments (including comment fields)

### Object Clause

The object is what the verb acts on. Some verbs take parameters and variables as objects and some take keywords; some take both; some might take no object clause. In the case of ADD, the keyword objects are: LINE, LINES, STMT and STMTS and for DEL, the operands are STMT, STMTS, COMMENT-LINE, and COMMENT-FIELD.

### Prepositional Clauses

The prepositional clause is one way to limit the scope of a change command. Some commands such as DEL do not require a preposition clause; others such as CHG or STRCHG do require a FROM-TO preposition clause. Whether the ADD command takes a prepositional phrase depends on whether you want to add a parameter/subparameter or a statement/line. The former does not require a prepositional phrase; the latter does.

### Delimiter

The END delimiter identifies the end of an ON command set; the ENDADD delimiter indicates the end of a group of lines or statements for an ADD STMT or ADD LINE command (ENDADD). Each of these delimiter images contains the delimiter beginning in column 1 and no other text or tokens. While the END at the end of the change command member is optional, JOB/SCAN requires an ENDADD for each ADD LINE or ADD STMT command; the end-of-file is not enough to end both an ADD clause and the ON selector.

The END delimiter terminates the scope of the previous selectors.

### Comments

You may include comments in a Change member using the delimiter /\*. There is no ending delimiter. To continue comments on another line, begin the comment with /\*. [Comments in a JCL Change Member](#) shows examples of different comment styles in a Change member.

## Comments in a JCL Change Member

```

/* THIS IS A FULL LINE COMMENT.  THE DELIMITER STARTS IN COL. 1
/* CONTINUING THE COMMENT ON THE NEXT LINE.
ON JOB
  CHG CLASS FROM A TO J  /*THIS IS A PARTIAL LINE COMMENT
  ADD TIME=1440          /*HERE IS ANOTHER PARTIAL LINE COMMENT
END                      /*END THE COMMAND SET

```

## Special Notation Used in this Topic

Some examples in this topic might contain these notations:

```

...
Change commands here
...

```

**Or**

```

...
JCL lines here
...

```

This notation indicates that the sample is a portion of a larger section of code and emphasizes the commands necessary to accomplish a particular change or the JCL sample that emphasizes a point.

## JCL Change Commands Order of Application

JOB/SCAN applies Change commands to a JCL member first on a JCL statement-by-statement basis, then in the order the commands appear in the change member. In a typical JCL member, JOB/SCAN first reads a JOB statement and applies all the ON JOB command sets to that statement. Next, JOB/SCAN might read a STEP (an EXEC) statement and apply all the ON EXEC (or as appropriate ON EXPGM or ON EXPROC) command sets. If DDs follow the STEP, JOB/SCAN then applies all the ON DD command sets. JOB/SCAN continues in this fashion until it has read and processed all the JCL statements. See [Example 1](#).

If there are multiple *ON* command sets for a JCL statement type, or if ON JCL or ON \* also appear, JOB/SCAN applies the commands in the order they appear in the JCL change member. See [Example 2](#).

### Example 1

In this example, JOB/SCAN applies the Change Commands to the input JCL in this order:

- ON JOB 2
- ON JOB 1
- ON EXEC
- ON DD

Input JCL	Commands
//ASGUSRB JOB XXXX, 'JOBZ', NOTIFY=USER1,	ON EXEC
// MSGLEVEL=1,	...
// MSGCLASS=X,	ON JOB /* ON JOB 2
// REGION=0M	...

Input JCL	Commands
//STEP1 EXEC MYPROG,DSN=D.E.F	ON DD
//PCFDEL DD DSN=ASGPCF.USER1,	...
// DCB=(BLKSIZE=27920),	ON JOB /* ON JOB 1
// UNIT=3390	...

## Example 2

Change Commands in ON JCL and ON \* sets are applied in the order they appear in the Change member relative to other ON sets. Compare Example 2.A with 2.B; each uses this input JCL:

---

```
//ASGUSRB JOB XXXX, 'JOBZ', NOTIFY=USER1,
//          MSGLEVEL=1,
//          MSGCLASS=X,
//          REGION=0M
...
```

---

### Example 2.A

JOB/SCAN applies these ON command sets in this order: \*, JOB, JCL.

---

```
ON *
  CHG NOTIFY FROM USER1 TO USER2
ON JOB
  CHG NOTIFY FROM USER2 TO USER3
ON JCL
  CHG NOTIFY FROM USER3 TO USER4
```

---

The JCL after:

---

```
//ASGUSRB JOB XXXX, 'JOBZ', NOTIFY=USER4,
//          MSGLEVEL=1,
//          MSGCLASS=X,
//          REGION=0M
...
```

---

Notice the value of the NOTIFY parameter is USER4, showing that JOB/SCAN applied all three change command sets.

### Example 2.B

JOB/SCAN applies these commands in this order: JOB, \*, JCL.

---

```
ON JOB
  CHG NOTIFY FROM USER2 TO USER3
ON *
  CHG NOTIFY FROM USER1 TO USER2
ON JCL
  CHG NOTIFY FROM USER3 TO USER4
```

---

The JCL after:

---

```
//ASGUSRB JOB XXXX, 'JOBZ', NOTIFY=USER2,
//          MSGLEVEL=1,
//          MSGCLASS=X,
//          REGION=0M
...
```

---



Notice the value of the NOTIFY parameter is USER2, showing that JOB/SCAN applied the value from the ON \* command set.

## Selector Clause (ON Command)

As stated in the General Syntax section, the ON command establishes the scope for the change commands which follow. The ON command is terminated by an END command or by the end of the member. Together, the ON and END commands surround a command set.

### Operands

There are several operands for the ON command and these can be divided into two groups: those with broad application and those that work on specific JCL statement types.

Operand	Applies to
*	Every type of statement.
JCL	Everything except control cards.
JCLONLY	Everything except control cards and JES cards.
NONJCL	Control cards only. Only STRCHG may be used with NONJCL.
NONJES	Everything except JES statements.
JES	All JES statements.
JES2	All JES2 statements.
JES3	All JES3 statements.
COMMENT	All comment statements.
EXPGM	EXEC statements that invoke programs.
EXPROC	EXEC statements that invoke PROCs.
\$COMMAND	All JES2 \$ commands.
*COMMAND	All JES3 * commands.
DELIMITR	JCL delimiter statements, usually beginning with the characters /*.

For JCL, the statement-specific types are: JOB, EXEC, DD, SET and so on; and for JES2 or JES3 JOBPARM, FORMAT, NOTIFY, and so on.

### Examples

Each of these command set examples gets a different scope from the ON command. Scope is important to understand,

because JOB/SCAN performs changes upon all statements that meet the criterion of the ON statement. For example adding a UNIT parameter to every DD statement uses these commands:

```
ON DD
ADD STMT
UNIT=NAME
END
```

Adding a comment line to every statement in the member uses these commands:

```
ON JCL
ADD STMT
/** ADD THIS TO EVERY LINE OF JCL
END
```

This third example deletes every statement in the member (all statements, control cards, comments, in short it creates an *empty* member, so use extra care when using ON \*):

```
ON *
DEL STMT
END
```

There are several ways to modify scope; these are discussed in [Setting the Scope](#).

## Verb Clause

The verb clause does the work of Change. This section presents the details of the Object and the Preposition clauses for each verb and describes how to tell Change precisely what to change and how to change it. The verbs for Change are:

ADD, DEL, CHG, STRCHG, ANNOTATE



ANNOTATE is discussed in [Advanced Features](#).

### ADD

The ADD verb clause specifies the insertion of variables, parameters, subparameters, statements, lines and comments. You may add any 1 to 80-byte image except the special ENDADD delimiter image.

Because ADD inserts JCL wherever you ask it, make sure that you use ADD STMT and ADD LINE appropriately and rerun JOB/SCAN validation on all newly changed JCL.

### Adding Variables, Parameters, and Subparameters

The commands to add single variables, parameters and subparameters use the same basic syntax. Do not use STMT, STMTS, LINE, LINES, nor any prepositions.

#### Syntax

- To add a variable:

```
ADD variable=value
```

- To add parameters and subparameters:

```
ADD parametername=value
ADD parametername:subparametername=value
```

- Each ADD command that adds a parameter or subparameter must have a *keyword* equated to a *keyword-value*. For example:

```
ADD USERVAR=ABCDEFGH
ADD CLASS=A
ADD DCB:DSORG=PO
```

- If the value of a variable, parameter, or subparameter contains spaces, enclose the string in double quotes:

```
ADD USERPARM="A B C"
```

## Adding Statements and Lines

JOB/SCAN can add statement and lines:

- as the first statement or line of a member
- as the last statement or line of a member
- directly after or before the statement or line that contains a *pattern*.

### Syntax

```
ADD STMT|STMTS { AT TOP|AT BOTTOM|AFTER pattern|BEFORE pattern}
new-statements-to-add-here
ENDADD
```

```
ADD LINE|LINES {AT TOP|AT BOTTOM|AFTER pattern|BEFORE pattern}
new-lines-to-add-here
ENDADD
```

For the ADD STMT verb clause with BEFORE or AFTER *pattern*, JOB/SCAN does not insert one statement into another; rather, when JOB/SCAN locates the *pattern*, it places the new statement before or after the statement in which it found the *pattern*. There is no functional difference between STMT and STMTS operands. A statement is considered to be all the lines between *pattern* and ENDADD.

```
ADD STMT AFTER IEFBR14
//SYSUT1*
// LABEL=(,NL)
ENDADD
```

For the ADD LINE verb clause with BEFORE or AFTER *pattern*, JOB/SCAN inserts one or more lines before or after the line where JOB/SCAN found the *pattern*. You can use ADD LINE to insert multiple parameters and subparameters into statements. There is no functional difference between LINE and LINES operands. JOB/SCAN adds all the text between *pattern* and ENDADD. For example, all the text between SYSUT1 and ENDADD.

```
ADD LINE AFTER SYSUT1
//SYSIN DD DSN=SBA.V7R3M0.UPDMASTER,
// DISP=(OLD,DELETE)
ENDADD
```



- The positioning (BEFORE or AFTER *pattern*) is relative to the active ON statement. The pattern search starts at the image which satisfied the ON (and the IF, when present), and proceeds to the next image of the same type or of a hierarchically higher type. For example, if an ADD STMT AFTER *pattern* is subordinate to an ON JOB, the search for pattern begins with the current JOB image and proceeds to the start of the next JOB statement, or the end of the member, whichever comes first.
- In another example, if the same ADD STMT AFTER *pattern* was subordinate to an EXEC, EXPGM, or EXPROC ON clause, JOB/SCAN starts its search with the current EXEC statement and proceeds to the start of the next EXEC statement, the start of the next JOB statement, or the end of the member, in that order,



whichever comes first. The full, pattern-search hierarchy for all ON statement operands is shown in this table:

For this ON clause	JOB/SCAN Pattern search terminated by finding next
*	end-of-member (eom) only
JOB	JOB, eom
EXEC	EXEC, JOB, eom
DD	DD, OUTPUT, EXEC, JOB, eom
OUTPUT	OUTPUT, DD, EXEC, JOB, eom
PROC	PROC, PEND, eom
PEND	PEND, PROC, eom
All others	Same-type record, eom

- Statements inserted with 'ADD STMT(S)/ADD LINE(S)' may NOT be manipulated through STRCHG, CHG, ADD, or DEL Change verbs in the same Change execution. Should these statements need to be manipulated, request that JOB/SCAN write the changed JCL, then make a subsequent run with different rules.
- The restriction on manipulation includes prohibition of tests against values of parameters in inserted statements and the checking for pattern is done against the original JCL only, not against any statements that Change inserted.
- With ON JOB as the active ON, the AT BOTTOM clause implies insertion immediately prior to the null ('/') statement, if any. This prevents inadvertently adding JCL between jobs. If lines need to be added between jobs, do so with the BEFORE or AFTER pattern syntax.
- JOB/SCAN processes ADD STMT(S) and ADD LINE(S) commands last after it processes all other change verbs. The IF condition, when present, must still be true after JOB/SCAN processes all STRCHG, CHG, DEL, and ADD commands in order for the ADD STMT(S)/ADD LINE(S) rules to take effect. For example these Change rules add a line and a UNIT parameter:

```
ON DD
IF UNIT=DASD
ADD STMT AFTER UNIT=DASD
/*ADD THIS LINE
ENDADD
ADD UNIT=TAPE
CHG UNIT FROM DASD TO TAPE
END
```



So, these commands change this JCL:

```
...
//PCFDEL DD DSN=ASGUSR.JCL,
//          UNIT=DASD,
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920)
...
```

to this (and not adding the comment because UNIT changed to TAPE):

```
...
//PCFDEL DD DSN=ASGUSR.JCL,
//          UNIT=TAPE,
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920)
...
```

You should note that DASD was changed to TAPE by the action of the CHG UNIT command. Now, the IF UNIT=DASD test is no longer true and JOB/SCAN does not perform the ADD STMT command. JOB/SCAN performs the CHG UNIT command before the ADD STMT command even though the CHG UNIT command is the last command in the member and outside the scope of the IF UNIT command.

If you comment out or remove the CHG UNIT rule:

```
...
/*CHG UNIT FROM DASD TO TAPE
...
```

the IF UNIT=DASD test remains true, JOB/SCAN employs the ADD STMT rule, and the JCL becomes:

```
...
//PCFDEL DD DSN=ASGUSR.JCL,
//          UNIT=DASD,
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920)
A/*ADD THIS LINE
...
```

## Adding Comments

You can add comment lines (Change does not add comment fields) to members with the ADD command. Use either STMT (which adds the comment after the statement containing *pattern*) or LINE (which adds the comment after the line containing *pattern*). For example:

---

```
ON JCL
IF PGM=IEFBR14
ADD STMT AFTER EXEC
/*
/** THESE LINES GO
/** AFTER EXEC
/** STATEMENT WHEN PROGRAM IS IEFBR14
/**
ENDADD
```

---

## DEL

The DEL verb clause specifies the removal of variables, parameters, subparameters, statements, comment lines and comment fields. The DEL command does not take a preposition clause. Use great care when deleting lines or parameters of JCL. Make sure you replace them with equivalents or that you know that defaults are acceptable.

## Deleting a Parameter or Subparameter

### Syntax

To delete a parameter or subparameter:

```
DEL parametername
DEL parametername:subparametername
```

To delete a parameter or subparameter with a specific value:

```
DEL parametername=parametervalue
DEL parametername:subparametername=subparametervalue
```

Deleting particular parameters or subparameters is a matter of coding them explicitly. For example these commands delete the UNIT or DCB regardless of their values or subparameters:

```
DEL UNIT
DEL DCB:BLKSIZE
```

You can direct Change to delete only when the parameter or subparameter has a specific value. For example:

```
DEL UNIT=SYSDA
DEL DCB:BLKSIZE=2170
```

Change will delete the UNIT parameter and its value only when SYSDA is the coded. The BLKSIZE subparameter and its value will be deleted only if the value for BLKSIZE is 2170. To delete only the parm value for a specific parm, STRCHG must be used.

In DEL verb clauses, a wildcard is implied if no value is coded after a parameter name. For example, these two commands are equivalent and delete every CLASS parameter encountered throughout the active scope:

```
DEL CLASS=*
DEL CLASS
```



Do not include commas or parentheses before or after the parameter being deleted; in this case, JCL Change handles punctuation automatically.

## Deleting Statements

### Syntax

```
DEL STMT
DEL STMTS
```

The verb clause DEL STMT deletes an entire statement, not merely a line. In fact there is no way with the DEL verb to delete only one line of JCL.



There is no functional difference between STMT and STMTS. For example, this command set deletes any DD statement named SYSUT7:

```
IF DD=SYSUT7
ON DD
DEL STMT
END
```

Remember, use the DEL command with care and caution. These commands:

```
ON *
DEL STMT
END
```

delete all statements, commands, and comments in the member; in short, everything.

When DEL STMT is combined with an evaluation of a condition and the condition is met, all statements found until that condition is no longer true will be deleted. For example, these commands

```
ON *
IF PARM1=BEAR
DEL STMT
END
```

delete everything in the following JCL from SET1 through STEP2, after which PARM1 is set to something other than BEAR (namely, LION).

---

```
...
//SET1    SET PARM1=BEAR
//STEP1   EXEC PGM=IEFBR14
//DD1     DD DSN=ASGUSR.WORK.CNTL,DISP=SHR
//STEP2   EXEC PGM=IEFBR14
//SET2    SET PARM1=LION
...
```

---

If you start the command set with ON SET instead of ON \*, then, only SET1 is deleted because Change is focused only on SET statements, and PARM1 equals BEAR.

## Deleting a Comment-Line

### Syntax

```
DEL COMMENT-LINE
```

For the DEL command, the comment lines removed are all the comment lines following a statement. For example, these commands:

```
ON EXEC
DEL COMMENT-LINE
END
```

remove the two comment lines following the EXEC statement:

```
//STEP1 EXEC PGM=SORT, REGION=512K
/** SORTING THE INPUT BY
/** ALPHANUMERIC ASCENDING ORDER
//STEPLIB DD DSN=ASG.#CNTL.TEST
```

leaving this:

```
//STEP1 EXEC PGM=SORT, REGION=512K
//STEPLIB DD DSN=ASG.#CNTL.TEST
```

## Deleting a Comment-Field

### Syntax

```
DEL COMMENT-FIELD
```

The DEL command deletes the comments on the same line as the JCL. These commands remove the comment field from the EXEC statement:

```
ON EXEC
DEL COMMENT-FIELD
END
```

in this JCL:

```
//STEP1 EXEC PGM=XYZ **THIS IS A COMMENT FIELD**
```

leaving this:

```
//STEP1 EXEC PGM=XYZ
```

## CHG

The CHG verb changes keyword-parameter values for user-defined and native keywords. The handling is different for JCL keyword and JES positional styles of parameters. The prepositions FROM and TO are used with CHG to identify the target of the change command and exactly how to change it.

### Syntax

```
CHG parmname FROM oldparametervalue TO newparametervalue
CHG parmname:subparmname FROM oldparametervalue TO newparametervalue
```

CHG searches for the keyword parameter specified and, if found, searches for the FROM value specified and if a match is found, changes the value to the specified TO value from the change command. For example, in

```
ON DD
CHG UNIT FROM SYSDA TO 3380
END
```

JOB/SCAN searches DD statements for the keyword UNIT and if it finds UNIT, if its value is SYSDA then JOB/SCAN processes the CHG command. If JOB/SCAN does not find either the keyword UNIT or the FROM value SYSDA, then it does not process the CHG command.

## STRCHG

The STRCHG verb changes, on a character-by-character basis, any string or pattern to a replacement string. The prepositions FROM and TO identify the string to be changed and exactly how to change it. It is a powerful feature the effects of which should be thoroughly tested prior to being put into production.

Put STRCHG verbs first in the command member. Do not use IF, AND, and OR selection logic nor MISSING tests with the STRCHG verb.

### Syntax

```
STRCHG FROM sourcestring TO targetstring
```

Enclose the *sourcestring* or *targetstring* in double quotes to search for and replace an exact string. For example this command set:

```
ON DD
STRCHG FROM "ASG.*.TEST" TO "ASG.*.PROD"
END
```

changes this JCL:

```
//      ASG.*.TEST
```

to:

```
//      ASG.*.PROD
```

Because of the double quotes, JOB/SCAN does not evaluate the asterisks in the command set as wildcards, but as characters.

Enclose the string in double quotes when either or both the *sourcestring* or *targetstring* contains spaces. For example this command set:

```
ON DD
STRCHG FROM UNIT=CART, TO "UNIT=SYSDA, SYSDA COMMENT"
END
```

changes this JCL:

```
//      UNIT=CART,
```

to:

```
//      UNIT=SYSDA, SYSDA COMMENT
```



You can use STRCHG to change parameter names or to change strings that span parameters or operators. For example, use STRCHG to change the values for the variables that begin with ENV in this JCL:

```
//MYPROC PROC ENV01=ABCENV,DB2VAR=P,
//          ENVDB2=P
```

The intent of this command set:

```
ON PROC
STRCHG FROM ENV*=P, TO ENV*=Y
STRCHG FROM DB2VAR=P TO DB2VAR=Q
END
```

is to change ENVDB2=P to ENVDB2=Y and DB2VAR=P to DB2VAR=Q. Change sees the strings ENVDB2=P and ENV,DB2VAR=P as satisfying the \* in the FROM condition and changes =P and =Q to =Y, the latter of which was not intended. In this case, reversing the order of the STRCHG commands produces the intended result.

The wildcard characters \* and % can be used in matched pairs to define *sourcestring* or *targetstring*, or they can be omitted. For example:

```
STRCHG FROM *DSN=TEST.PAYROLL* TO *DSN=PROD.PAYROLL*
STRCHG FROM %DSN=TEST.PAYROLL% TO %DSN=PROD.PAYROLL%
STRCHG FROM DSN=TEST.PAYROLL TO DSN=PROD.PAYROLL
```

Change keeps the remainder of the strings to the left of DSN and to the right of PAYROLL the same in all cases.



- Do not use single quotes with the STRCHG verb.
- Do not mix double quotes with wildcards to delimit *sourcestring* or *targetstring*; use all one or all the other in each STRCHG command.

## Setting the Scope

While change commands work within the scope set by the ON command, there are a few ways to modify that scope: conditions, the use of subparameters, wildcards and missing tests. MISSING tests are discussed in [Advanced Features](#).

### Conditional Commands

On those verbs that take a preposition, namely the ADD, CHG and STRCHG, the preposition clause further limits the scope. However, the selectors IF, AND, and OR also assist the ON selector in limiting the scope. These commands set up conditions under which the commands within the ON command set are restricted. These commands take several special operands listed in this table.

Operand	Applies to
ACCT=	The accounting information from the JOB statement.
DD=	The ddname from the DD statement.
DDCONCAT=	Only the named DD statement in a concatenated DD.
JOB=	The job name from the JOB statement.
NAMEFIELD=	The name field from the JOB, EXEC, PROC, DD, or OUTPUT statement.

Operand	Applies to
PGMRNAME=	The programmer name from the JOB statement. Put the value for this operand in quotes: PGMRNAME='DEPT10'
PROC=	The PROC name from the EXEC statement.
PROCNAME=	The name field from the PROC statement.
STEP=	The step name from the EXEC statement.
USERID=	The USERID from the JES2 NOTIFY statement.

### Simple Conditional Tests

For instance, the command set:

```
ON DD
IF PGM=IEFBR14
ADD UNIT=DASD
END
```

add the UNIT parameter to all DD statements after a step invoking IEFBR14 is located.

IF, AND and OR selectors may also come before the ON command:

```
IF PGM=IEFBR14
ON DD
ADD UNIT=DASD
END
```

There is no difference between the two styles. In each case, the IF selector only applies to the END-delimited ON command set it is located before or within.

### Complex Conditional Tests

Complex tests can be written using AND and OR selectors in their usual connotations. AND adds an additional mandatory condition that must be met before the commands take place. In this example, UNIT is added only if the conditions of PGM and UNIT are both met:

```
ON DD
IF PGM=IEFBR14 AND UNIT=DASD
ADD UNIT=DASD
END
```

The OR test, on the other hand, specifies alternate conditions. With these commands:

```
ON DD
IF PGM=IEFBR14 OR UNIT=DASD
ADD UNIT=DASD
END
```

Change adds UNIT if either the PGM or UNIT condition is met.

## Applying Conditional Tests Across Multiple ON Sets

Generally, an IF test applies to the ON set it is in or before, up to the next explicit END delimiter or end of member. It is possible for the scope of an IF selector to encompass multiple ON command sets by omitting explicit END delimiters between ON sets:

```
IF MSGCLASS=X
ON JOB
  CHG NOTIFY FROM USER1 TO USER2
ON DD
  CHG UNIT FROM 3390 TO DASD
ON EXEC
  CHG DSN FROM A.B.C TO D.E.F
END
```

So, JOB/SCAN applies each CHG command to each statement type only if MSGCLASS=X.

Use care when omitting the END delimiters, particularly if you test a parameter value that you subsequently change. JOB/SCAN adjusts the result of the test to follow. For example:

```
IF NOTIFY=USER1
ON JOB
  CHG NOTIFY FROM USER1 TO USER2
ON DD
  CHG UNIT FROM 3390 TO DASD
END
```

If the IF NOTIFY test is true, the ON JOB change command changes the NOTIFY parameter value to USER2. Consequently, the IF test is now false. This, combined with the omission of the END delimiter between ON JOB and ON DD, prevents JOB/SCAN from applying the ON DD change command.



- The scope of an IF selector is terminated by the next explicitly coded END delimiter.
- The operands of IF, AND, and OR selectors can specify either equal (=) or not equal (≠). For example:

```
IF PGM≠IDCAMS
```

For most terminals and emulator programs in the United States the "not" character (≠) is Shift 6. If this key combination is not the EBCDIC "x5F" *not-sign* character, consult the emulator program to locate this character.

- If there are embedded blanks or special characters, the IF command operand needs to be enclosed in double quotation marks.
- Parentheses to any level of nesting may be used to alter the default grouping of AND and OR selectors. If parentheses are omitted, AND takes precedence over OR.
- Only keyword subparameters can be specified using the subparameter syntax. To specify positional subparameters, use CHG or STRCHG. For example:

```
IF DISP=" (NEW,KEEP,KEEP) "
  CHG DISP FROM * TO " (OLD,KEEP,KEEP) "
END
```



- See [Modifying JCL](#) for examples of how to use DD and DDConcat operands for the IF statement.

## Parameters and Subparameters

Parameters and subparameters can be specified on Change verb statements and conditional tests to limit scope. Given this JCL,

---

```
//STEP1 EXEC PGM=IEFBR14
//PCFDEL DD DSN=ASGUSR.JCL,
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920),
//          DISP=(SHR,DELETE)
```

---

these Change commands test the program parameter on the EXEC statement and change the BLKSIZE subparameter of the DCB parameter to 8096 if it is 27920:

```
ON DD
IF PGM=IEFBR14
CHG DCB:BLKSIZE FROM 27920 TO 8096
END
```

JOB/SCAN changes the JCL to:

---

```
//STEP1 EXEC PGM=IEFBR14
//PCFDEL DD DSN=ASGUSR.JCL,
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=8096),
//          DISP=(SHR,DELETE)
```

---

Parameters and subparameters may also be specified on conditions. These commands test the RECFM subparameter on the DCB parameter then make the same change to the BLKSIZE subparameter:

```
IF DCB:RECFM=FB
CHG DCB:BLKSIZE FROM 27920 TO 8096
```

## Wildcards

Wildcard characters allow commands that key on character string matches to key on *patterns* of characters in addition to precise matches.

### Percent (%)

The percent sign (%) wildcard character matches one character at a specific position in a string. For example, to test if the third character of a job name is P, code:

```
IF JOB=%P*
```

### Asterisk (\*)

The asterisk (\*) wildcard character matches any string of characters of any length. For example, the wildcard in this Change command segment changes any string of characters beginning with SYS to SYSDA:

```
FROM SYS* TO SYSDA
```

### To change an asterisk in JCL

- Enclose the string it comes in with double quotes.

```
CHG  SYSOUT FROM "*" TO OUTDSN
```

### To change this dataset name:

```
ACCTNG.PAYROLL.TEST.SALARY
```

use this command:

```
CHG DSN FROM *.*.TEST.* TO *.*.PROD.*
```

It changes the dataset name to:

```
ACCTNG.PAYROLL.PROD.SALARY
```

### Tilde (~)

The tilde (~) wildcard character is used in fields to drop characters matched by the asterisk (\*). For example, suppose you want to drop one node and rename another in the dataset `ACCTNG.PAYROLL.TEST.SALARY`. You could try these commands, dropping the second pair of \*. and changing TEST to PROD in the TO clause:

```
ON DD STRCHG FROM *.*.TEST.*
      TO *.PROD.*
END
```

Verification fails on these commands because JOB/SCAN requires the same number of wildcards on the TO clause as were on the FROM clause. To correct this, insert a ~ where the \*. for the second node had been:

```
ON DD STRCHG FROM *.*.TEST.*
      TO *.*~PROD.*
END
```

These commands pass verification and JOB/SCAN changes the dataset name to `ACCTNG.PROD.SALARY`.

### Special Wildcard Rules

- In DEL verbs, a wildcard is implied if no value is coded after a parameter name. For example, these two statements are equivalent:

```
DEL CLASS=*
DEL CLASS
```

- Wildcard characters may be used to define the limits of the STRCHG verb, or they can be omitted. For example, these two specifications are equivalent:

```
*DSN=MY.PROD.*
DSN=MY.PROD.
```

This is because STRCHG will change only the string specified in the FROM clause; text strings before and after the specified string are left as is.

## Nuances of Scope

Because of the way JCL can be coded and the way JOB/SCAN parses JCL statements, the scope for a particular Change command can be nuanced. For instance, this JCL defines a UNIT parameter for PROC01 procedure:

---

```
//ASGUSRB JOB XXXX, 'JOBZ', NOTIFY=ASGUSR
//PROC01 PROC UNIT=2295
//PSTEP01 EXEC PGM=PROG01, PARM='&UNIT'
//      PEND
//STEP1 EXEC PROC01, UNIT=2295
//PCFDEL DD DSN=ASGPCF.USER1,
//      DCB=(BLKSIZE=27920)
//OUTPUT DD DSN=ASGUSR.OUTPUT
```

---

These Change commands which normally would change BLKSIZE if UNIT is missing, do not change any lines in the JCL even though there is no UNIT on the DD statement:

```
ON DD
IF UNIT IS MISSING
CHG DCB:BLKSIZE FROM 27920 TO 28970
END
```

The UNIT parameter on STEP1 EXEC PROC01 causes the MISSING tests for UNIT on the DDs for STEP1 to test FALSE (UNIT not missing) and BLKSIZE is not changed.

JOB/SCAN restarts scanning on a statement-by-statement basis, resetting values for parameters, subparameters and variables. JOB/SCAN restarts:

- DD scanning for each new DD encountered (and other statement types when new statements are encountered)
- EXEC or STEP scanning (and all associated DD and other statement types) for each new step encountered
- JOB scanning (and all associated steps, DD and other statements types) for each new JOB encountered.

Also, tests for previously defined parameters, such as NOTIFY in the JOB statement, can be used to determine whether an action takes place.

## Basic Change Examples

This section provides many examples showing how these commands may be combined. Make sure you verify your commands with JOB/SCAN prior to running them. See [Advanced Features](#) for other tests and member annotation.

### Changing JOB Card Parameters

To change the accounting field in a JOB statement use these commands:

```
ON JOB
IF PGMNAME='ASGUSRB'
CHG ACCT FROM XXXX TO DEPT10
END
```

The JCL before:

```
//USRJB1 JOB XXXX, 'JOBZ', NOTIFY=ASGUSRB,
//      MSGLEVEL=1,
//      MSGCLASS=X, CLASS=F,
//      REGION=0M
```

The JCL after:

```
//USRJB1 JOB DEPT10,'JOBZ',NOTIFY=ASGUSRB,
//          MSGLEVEL=1,
//          MSGCLASS=X,CLASS=F,
//          REGION=0M
```

### Changing DD Statement Parameters

To change the UNIT parameter in a DD statement use these commands:

```
ON DD
IF NAMEFIELD=PCFDEL
CHG UNIT FROM DASD TO 3390
END
```

The JCL before:

```
//PCFDEL DD DSN=ASGUSR.JCL,
//          UNIT=DASD,
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920)
```

The JCL after:

```
//PCFDEL DD DSN=ASGUSR.JCL,
//          UNIT=3390,
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920)
```

### Changing Strings (STRCHG)

To change the character string DASD to 3390 wherever it occurs use these commands:

```
ON *
STRCHG FROM DASD TO 3390
END
```

Consider this DD statement and its comment field:

```
//PCFDEL DD DSN=ASGUSR.JCL,
//          UNIT=DASD,          UNIT IS DASD
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920)
```

The resulting JCL:

```
//PCFDEL DD DSN=ASGUSR.JCL,
//          UNIT=3390,          UNIT IS 3390
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920)
```

### Changing NON-JCL

In this example, JOB/SCAN changes ASG. to LSP.:

```
ON NONJCL
STRCHG FROM ASG. TO LSP.
END
```

The JCL before:

```
DEFINE CLUSTER (NAME (ASG.PRD)
```

The JCL after:

```
DEFINE CLUSTER (NAME (LSP.PRD)
```

In other lines of JCL, if ASG. were to appear in a DSN parameter, JOB/SCAN would not change it by this rule.

### Changing JES2 Statements

To change the BYTES parameter on a JOBPARM statement:

```
ON JOBPARM
CHG BYTES FROM * TO 32767
```

The JCL before:

```
/*JOBPARM BYTES=80
```

The JCL after:

```
/*JOBPARM BYTES=32767
```

To change the printer name on a ROUTE statement:

```
ON ROUTE
STRCHG FROM ASG04 TO RMT07
```

The JCL before:

```
/*ROUTE PRINT ASG04
```

The JCL after:

```
/*ROUTE PRINT RMT07
```

To change a string on any JES2 statement in a member:

```
ON JES2
STRCHG FROM FRED TO BILL
```

The JCL before: (The JES/2 output statement must be referred to as JES2.)

```
/*OUTPUT UCS=FRED
```

The JCL after:

```
/*OUTPUT UCS=BILL
```



## Adding a Subparameter

To add a subparameter specify the keyword and the keywordvalue in this case DSORG=PO:

```
ADD DCB:DSORG=PO
```

The JCL before:

```
//DD1 DD DCB=(LRECL=80),...
```

The JCL after:

```
//DD1 DD DCB=(LRECL=80,DSORG=PO),...
```

When adding a subparameter, JOB/SCAN does not add the subparameter if a subparameter of the same name already exists.

## Adding a Parameter

To add a parameter specify the keyword and the keywordvalue in this case UNIT=3390:

```
ON DD
IF UNIT IS MISSING
ADD UNIT=3390
END
```

The JCL before:

```
//PCFDEL DD DSN=ASGUSR.JCL,
//          DCB=(LRECL=80,RECFM=FB),
//          DISP=(SHR,DELETE)
```

The JCL after:

```
//PCFDEL DD DSN=ASGUSR.JCL,
//          DCB=(LRECL=80,RECFM=FB),
//          DISP=(SHR,DELETE),
//          UNIT=3390
```

Notice that JOB/SCAN added the new parameter at the end of the DD statement, adding a comma after the DISP parameter.

## Adding a Statement

### To add an entire statement

1. Identify its position: after or before a given statement or, at the top or bottom of a member.
2. Then use commands similar these to add a new DD after the statement containing BLKSIZE=27920:

```
ON DD
ADD STMT AFTER BLKSIZE=27920
//NEWDD DD DSN=ASGUSR.NODE1.NODE2
ENDADD
```

The JCL before:

```
//PCFDEL DD DSN=ASGUSR.JCL,
```

```
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920),
//          DISP=(SHR,DELETE)
```

The JCL after:

```
//PCFDEL DD DSN=ASGUSR.JCL,
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920),
//          DISP=(SHR,DELETE)
//NEWDD DD DSN=ASGUSR.NODE1.NODE2
```

Notice that JOB/SCAN added the NEWDD after the DISP parameter of the PCFDEL DD, and not in the middle of it. Contrast this with [Adding a Line](#).

### Adding a Line

#### To add a line

1. Identify its position: after or before a given line (containing a searched-for pattern) or, at the top or bottom of a member.
2. Use similar commands to add a line after a line containing BLKSIZE=27920:

```
ON DD
ADD LINE AFTER BLKSIZE=27920
//          UNIT=3390    UNIT IS 3390
ENDADD
END
```

The JCL before:

```
//PCFDEL DD DSN=ASGUSR.JCL,
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920),
//          DISP=(SHR,DELETE)
```

The JCL after:

```
//PCFDEL DD DSN=ASGUSR.JCL,
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920),
//          UNIT=3390    UNIT IS 3390
//          DISP=(SHR,DELETE)
```



JOB/SCAN added the UNIT line after the BLKSIZE line, but because the line in the Change commands did not include a comma, JOB/SCAN did not add one. In this case JOB/SCAN added just what you told it and nothing more. If you modify the Change commands in this section to:

```
ON DD
IF DCB:BLKSIZE=27920
ADD UNIT=3390
END
```



JOB/SCAN not only handles the punctuation with a comma on the DISP parameter, it also inserts the // delimiter at the head of the line it adds for the UNIT parameter:

```
//PCFDEL DD DSN=ASGUSR.JCL,
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920),
//          DISP=(SHR,DELETE),
//          UNIT=3390
```

### Adding Comments

Adding a comment is the same as adding a statement or line; make sure the lines added conform to the rules for JCL comments. JOB/SCAN does not add comment fields.

To add a comment, first identify its position: after or before a given statement, at the top or bottom of a job. Then, use commands similar these:

---

```
ON DD
ADD LINE AFTER BLKSIZE=27920
/* USE BLKSIZE 27920 FOR ALL INPUT
/* DATA SETS FOR PGM MYPGM
ENDADD
END
```

---

JCL before:

---

```
//PCFDEL DD DSN=ASGUSR.JCL,
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920),
//          DISP=(SHR,DELETE)
...
```

---

JCL after:

---

```
//PCFDEL DD DSN=ASGUSR.JCL,
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920),
/* USE BLKSIZE 27920 FOR ALL INPUT
/* DATA SETS FOR PGM MYPGM
//          DISP=(SHR,DELETE)
...
```

---



The ADD LINE command put the comments in the middle of the DD statement. You may also use the ADD STMT form to add comments. Commands in this form puts the comments after the DD:

```
ON DD
ADD STMT AFTER BLKSIZE=27920
/* USE BLKSIZE 27920 FOR ALL INPUT
/* DATA SETS FOR PGM MYPGM
ENDADD
```



END

**JCL after:**

```
//PCFDEL DD DSN=ASGUSR.JCL,
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920),
//          DISP=(SHR,DELETE)
//* USE BLKSIZE 27920 FOR ALL INPUT
//* DATA SETS FOR PGM MYPGM
...

```

**Modifying JCL**

Use the DD and DD CONCAT parameters when you want to consider all the DDs in a named DD statement or solely the first one. Consider this JCL for these examples:

```
//ASGUSRB JOB XXXX, 'JOBZ', NOTIFY=ASGUSR,
//          MSGCLASS=X, CLASS=F,
//          REGION=0M
//STEP1 EXEC PGM=MYPROG
//SYSEXEC DD DISP=SHR, DSN=ASGCO.@UTL.EXEC
//          DD DISP=SHR, DSN=ASGUSR.USU.EXEC

```

**Example 1**

Use commands similar to these to add a parameter to each DD in the SYSEXEC DD:

```
ON DD
IF DD=SYSEXEC
ADD UNIT=DASD
END

```

The JCL becomes:

---

```
//ASGUSRB JOB XXXX, 'JOBZ', NOTIFY=ASGUSR,
//          MSGCLASS=X, CLASS=F,
//          REGION=0M
//STEP1 EXEC PGM=MYPROG
$/SYSEXEC DD DISP=SHR, DSN=ASGCO.@UTL.EXEC,
$/          UNIT=DASD
$/          DD DISP=SHR, DSN=ASGUSR.USU.EXEC,
$/          UNIT=DASD

```

---

**Example 2**

Use commands similar to these to add a parameter to the first SYSEXEC DD only:

```
ON DD
IF DD CONCAT=SYSEXEC
ADD VOL=SER=VOLSER
END

```

The JCL becomes:

---

```
//ASGUSRB JOB XXXX, 'JOBZ', NOTIFY=ASGUSR,
//          MSGCLASS=X, CLASS=F,
//          REGION=0M
//STEP1   EXEC PGM=MYPROG
$/SYSEXEC DD DISP=SHR, DSN=ASGCO.@UTL.EXEC,
$/          VOL=SER=VOLSER
$/          DD DISP=SHR, DSN=ASGUSR.USU.EXEC,
```

---

## Deleting Statements

To delete a statement, first identify it with the various tests available to Change. Choose the test carefully so that JOB/SCAN deletes only the statement that needs to go.

Then use commands similar to these to remove an entire step:

---

```
IF DCB:BLKSIZE=27920
ON DD
DEL STMT
END
ON EXEC IF PGM=MYPROG
DEL STMT
END
```

---

JCL before:

---

```
//ASGUSRB JOB XXXX, 'JOBZ', NOTIFY=ASGUSR,
//          MSGLEVEL=1,
//          MSGCLASS=X, CLASS=F,
//          REGION=0M
//STEP1   EXEC PGM=MYPROG
//PCFDEL  DD DSN=ASGUSR.JCL,
//          DCB=(LRECL=80, RECFM=FB,
//          BLKSIZE=27920),
//          DISP=(SHR, DELETE)
//STEP02  EXEC YOURPGM
...

```

---

JCL after (all of STEP1 is removed):

---

```
//ASGUSRB JOB XXXX, 'JOBZ', NOTIFY=ASGUSR,
//          MSGLEVEL=1,
//          MSGCLASS=X, CLASS=F,
//          REGION=0M
//STEP02  EXEC YOURPGM
...

```

---

## Deleting a Parameter or Subparameter

To delete the UNIT parameter regardless of its value, use commands similar to these:

```
ON DD
IF NAMEFIELD=PCFDEL
DEL UNIT
END
```

JCL before:

```
//PCFDEL  DD DSN=ASGUSR.JCL,
//          DCB=(LRECL=80, RECFM=FB,
//          BLKSIZE=27920),
//          UNIT=3390,
//          DISP=(SHR, DELETE)
```

JCL after:

```
//PCFDEL DD DSN=ASGUSR.JCL,
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920),
//          DISP=(SHR,DELETE)
```

To delete a UNIT parameter with a specific value, use commands similar to these:

```
ON DD
DEL UNIT=3370
END
```

on this JCL:

```
//PCFDEL DD DSN=ASGUSR.JCL,
//          DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920),
//          UNIT=3380,
//          DISP=(SHR,DELETE)
```

JOB/SCAN does not delete the UNIT parameter because its value is not 3370.

### Deleting Comment Fields

JOB/SCAN can delete all the comment fields in a member or only those on a specific statement. To delete comment fields on a specific statement, use commands similar to these:

```
IF DCB:BLKSIZE=27920
ON DD
DEL COMMENT-FIELD
END
```

JCL before:

```
//PCFDEL DD DSN=ASGUSR.JCL,          <== PUTNAME HERE
//          DCB=(LRECL=80,RECFM=FB, <== LRECL HERE
//          BLKSIZE=27920),
//          UNIT=3390,              <== CHANGE UNIT
//          DISP=(SHR,DELETE)
```

JCL after:

```
//PCFDEL DD DSN=ASGUSR.JCL,DCB=(LRECL=80,RECFM=FB,
//          BLKSIZE=27920),
//          UNIT=3390,DISP=(SHR,DELETE)
```



JOB/SCAN reformats the parameters according to Control table settings, so the results you get might be different than what this example shows.

To delete comment fields throughout an entire member, use commands similar to these:

```
ON JCL
DEL COMMENT-FIELD
```

```
END
```

### Deleting a Comment Line

JOB/SCAN deletes comment lines that come between statements. To remove such comment lines from JCL use commands similar to these:

```
IF DCB:BLKSIZE=27920
ON DD
DEL COMMENT-LINE
END
```

JCL before:

---

```
//PCFDEL DD DSN=ASGUSR.JCL,
//          DCB=(LRECL=80,RECFM=FB,
//* USE LRECL=80 FOR ALL DATA SETS
//          BLKSIZE=27920),
//          DISP=(SHR,DELETE)
//* USE BLKSIZE 27920 FOR MYPROG
//* AND USE DISP = SHR,DELETE
//NEWDD DD DSN=MY.DATA.SET
```

---

JCL after:

---

```
//PCFDEL DD DSN=ASGUSR.JCL,
//          DCB=(LRECL=80,RECFM=FB,
//* USE LRECL=80 FOR ALL DATA SETS
//          BLKSIZE=27920),
//          DISP=(SHR,DELETE)
//NEWDD DD DSN=MY.DATA.SET
```

---



With these commands, JOB/SCAN does not remove imbedded comments such as the one just after the DCB parameter:

```
//* USE LRECL=80 FOR ALL DATA SETS
```

To delete comment lines throughout an entire member, use commands similar to these:

```
ON JCL
DEL COMMENT-LINE
END
```

### Using IF

The IF test may come before the first ON selector in a Change member. This positioning of the IF test sets the scope for all ON selectors in the member.

```
IF DCB:BLKSIZE=27920
ON DD
```

When the IF test comes after an ON selector, only the scope of the ON selector is modified by the IF test.

```
ON DD
IF DCB:BLKSIZE=27920
```

When you use a subparameter in an IF test, then use the parameter:subparameter notation. These commands:

```
ON DD
IF DCB:BLKSIZE=27920
DEL UNIT
END
```

will only delete the UNIT parameter when the BLKSIZE equals 27920.

### Using OR

IF tests to set the scope of change commands maybe further refined by the OR selector.

```
ON DD
IF DCB:BLKSIZE=27920 OR UNIT=3390
DEL UNIT
END
```

### Using AND

IF tests to set the scope of change commands may be further refined by the AND selector.

```
ON DD
IF DCB:BLKSIZE=27920 AND UNIT=3390
DEL UNIT
END
```

### Using Wildcards

If you need to change a DCB subparameter, for example, and your site requires that all DCB:RECFM subparameters be FB, then use Change commands similar to these:

```
ON DD
IF DCB:RECFM IS NOT MISSING
CHG DCB:RECFM FROM * TO FB
END
```

The CHG command changes the value of the RECFM subparameter from whatever it is to FB, even if it was already FB. This technique is useful when you know a parameter is present and have no need to test its value. In the case of unique parameters and subparameters, JOB/SCAN does not provide a MISSING test for its value. Some more examples:

- The A\*M\* mask matches AM, ARM, ABCM, ABCMR, AMUTL, etc.
- The JOB=P%%DAY Change command phrase matches all jobs with a name that consists of the character P followed by *exactly two* other characters, then the characters DAY: PAYDAY, PZZDAY, P\$XDAY.
- The JOB=P\*DAY Change command phrase matches all jobs with a name that starts with P and ends with DAY: PAYDAY, PDAY, P\$DAY.
- The JOB=P\*%%Y Change command phrase matches all jobs with a name that consists of the character P followed by at least two other characters, then the character Y: PDAY, PAYDAY, PZZY, PDAZZY.

For more about using wildcards, see [Wildcards](#).



## Running JCL Change

You can run Change from the ISPF panels, from batch, or from ISPF Edit with an ASG-supplied Edit macro.

### ISPF (Foreground) Mode

JCL Change runs in foreground in any of these ways:

- **Verify mode.** In verify mode, JOB/SCAN verifies JCL Change commands to ensure that they are free of syntax errors.
- **Simulation (test) mode.** In simulation mode, JCL Change produces an output listing showing the complete JCL member as it would be with changes applied.
- **Write mode.** In write mode, JCL Change writes the changed JCL to an output member.

If you use the Change facility in the order Verify, Simulate, and Write you will efficiently employ JCL Change through JOB/SCAN's ISPF panels.

### Verify Mode

Use the V (verify) option from the command line on the first Change/Reformat JCL--Input panel.

Because JCL Change commands do not execute if the command syntax is incorrect, always verify any new Change commands before applying them to a JCL member to make sure they are free of JCL Change syntax errors.



- Verifying JCL Change commands can only be done in foreground mode or Edit macro mode and not in batch.
- Verified commands are only syntax-error-free and is no guarantee of desired operation; test the commands extensively to ensure that they do what you want them to do.

### *To verify the syntax of Change commands*

1. On the Change/Reformat JCL--Input panel, type V on the command line. Make sure the correct command library and member name are in the appropriate fields, and press Enter.

If JOB/SCAN finds errors, it displays a listing of the Change commands with the errors flagged.

2. Examine the errors detected by the verification step.
3. Press End to return to the Change/Reformat JCL--Input panel.
4. Correct any errors by typing M on the command line and press Enter.
5. Make corrections and press End.
6. Repeat steps 1 through 5 until no more errors are found.

### Simulation Mode

JCL Change simulation mode allows testing of JCL Change commands before changing a JCL member. The resulting output listing shows the complete JCL member as it would be when changed. The changed statements are flagged with a JCL change replacement character, typically a \$ (dollar sign) in column 1.



The Product Administrator may change the default JCL Change Replacement Character (\$) to a different character.

### ***To simulate a change***

1. From the JOB/SCAN Primary Menu, select the Change/Reformat JCL option.

The first Change/Reformat JCL panel appears.

2. Type F on the command line.
3. Type the name of the command library and member in the Command Lib and Member Name fields.

Obtain a directory of the command library by leaving the Member Name field blank.

4. Fill in the Input Library, Input Type, and Library Type fields. Use the Select Members and Exclude Members fields to specify the input JCL members, or leave them blank to display the directory. See [Selecting Members](#).
5. Type Y in the Change JCL field.
6. Press Enter.
7. On the second Change/Reformat JCL panel, N is the default in the Write Changed JCL? field. Leave the N in this field.
8. If the F option was selected in Step 2, type O (for output JCL report) in the JCL Change Report Display field and N in the JCL Change Report Hardcopy field.
9. Press Enter.

When the simulation is complete, JOB/SCAN displays an output listing ([Writing Changed JCL to an Output Member](#)).

### **Write Mode**

When you are satisfied that the JCL Change commands will produce the desired results, write the members to a new dataset.

### ***To create an updated JCL member***

1. From the JOB/SCAN Primary Menu, select the Change/Reformat JCL option. The Input Change/Reformat JCL panel appears.
2. Type F on the command line.
3. Type the name of the command library and member in the Command Lib and Member Name fields.

Obtain a directory of the command library by leaving the Member Name field blank.

4. Fill in the Input Library, Input Type, and Library Type fields. Use the Select Members and Exclude Members fields to specify the input JCL members, or leave them blank to display the directory. See [Selecting Members](#).
5. Type Y in the Change JCL field.
6. Press Enter.

7. On the Output Change/Reformat JCL panel, type Y (Yes) or C (ChgOnly) in the Write Changed JCL? field. To write changed JCL to an output member, type Y in the Write Changed JCL? field.
8. In the Output Lib field, type the name of an existing library to receive the changed JCL. ASG strongly recommends that a library other than the input library be used.
9. Specify the type of output library in the Library type field. Valid values are SEQ, PDS, PAN, LIB, and LIB-3.
10. Specify the types of JCL Change reports desired.

#### Writing Changed JCL to an Output Member

```

----- JOB/SCAN - CHANGE / REFORMAT JCL -----
COMMAND ==>

Write Changed JCL? ==> Y          (Yes/No/ChgOnly)

Output Lib      ==> WORK.CNTL
Library type    ==> PDS          (PDS, PAN, LIB, LIB-3, or SEQ)

JCL Change Report:
Display        ==> O (OutJCL/Audit/ChgOnly/None)
Hardcopy       ==> N (OutJCL/Audit/ChgOnly/None)

Processing Summary:
Display        ==> Y (Y/N)
Hardcopy       ==> N (Y/N)

Press <ENTER> to continue.

Press <END> to cancel.

```

11. Press Enter.



- ASG *strongly* recommends against overwriting the input JCL with the output JCL. Instead, write the output JCL to a different library.
- Consult the panel help for more information about the several fields.
- JOB/SCAN will not write to a LIB-3 dataset name that has the archive bit on. Turning off the archive bit will compromise the history for the member. Exercise care when working with archived files.

#### ISPF Background Mode

If a change is run against a particularly large library, run it in batch from the panels, or submit a batch job using the SJSSAMP library member JSMVS.

#### **To start JCL Change from the panels to run in background mode instead of in the foreground**

1. Follow the procedure given for foreground execution, but change the execution option from F (foreground) to B (background) on the command line of the first Change/Reformat JCL panel.

For more information, see [Running JCL Change](#) to choose a mode of operation.)

2. On the bottom of the first Change/Reformat JCL panel, the Change JCL, Reformat JCL, and Validate JCL fields must each have a value, either Y for yes, or N for no, and at least one must be Y.

- If all three are set to Y, the execution sequence is: change the JCL first, reformat it second, and validate the changed and reformatted JCL last.



- The selection characters \* and % are not allowed in the Member Name field.
- Change commands cannot be verified in batch. Instead, create and test commands using the Change/Reformat JCL panels in ISPF, or use the JVER and JCHG Edit macros to create and test the commands. Then execute the Change function in batch.

### Running JCL Change in Batch Using JSMVS

The SJSSAMP library contains two members for running JCL Change in batch mode:

- JSMVS — executable JCL to run JCL Change in batch.
- JS6PROC — the PROC invoked by JSMVS.

The SJSSAMP library member JSMVS contains the executable JCL to run JCL Change in batch mode. JSMVS executes the PROC named JS6PROC. The below figure shows a listing of the JSMVS member:

#### Sample JSMVS Member in SJSSAMP

```
000001 //JSMVS JOB (ACCT),'PROGRAMMER NAME'
000002 //*****
000003 //* EXECUTE JCL CHANGE (MVS SYSTEMS)
000004 //* REF: SAMPLIB(JSMVS)
000005 //*
000006 //* NOTE: MODIFY JOB STATEMENT AND ITEMS INDICATED BY ARROWS <==
000007 //*****
000008 // EXEC JS6PROC,PRM=C
000009 //JLCHMO DD DISP=SHR,DSN=YOUR.OUTPUT.PDS <== WRITE CHANGED JCL
000010 //JLRULES DD DISP=SHR,DSN=YOUR.CHANGE.CMDS (MBR) <== JCL CHANGE CMDS
000011 //SYSIN DD *
000012 INPUT SELECTION
000013 DSN=YOUR.JCLLIB <== INPUT JCL LIBRARY
000014 SELECT=YOURJOB <== MEMBER TO CHANGE
000015 SOURCE TYPE=PDS <== TYPE OF INPUT LIB
000016 //
```

#### To run JCL Change in batch mode

- Make a copy of and Edit the SJSSAMP library member JSMVS as follows. (See [Managing JOB/SCAN through Options Members](#) for more information.)
- Modify the JOB statement according to site standards.
- Change PRM=S to PRM=C in the EXEC statement.
- Add and Edit the JLRULES DD statement to specify the library and specific member containing the commands to use for this JCL Change run.
- Add and Edit the JLCHMO DD statement to specify the library where the changed JCL should be written. To perform a simulation, comment out this statement.
- Edit the Input Selection DSN= statement to specify the library containing the JCL members to change.
- Edit the SELECT= statement to specify the members to change.

8. Edit the SOURCE TYPE= statement to identify the type of library containing the JCL.
9. Submit JSMVS for execution.



If the name of JS6PROC has been changed, use the JOB/SCAN PROC Run Option to specify the new name of the PROC. See [JOB/SCAN PROC \(JSP\)](#) for complete information about Run Options.

### Running Change from the Edit Macros

There are two Edit macros to use with Change: JCHG to change JCL and JVER to verify change commands prior to running them.

### Changing JCL with JCHG

The JCHG Edit macro invokes JCL Change and/or Reformatter from within ISPF Edit, allowing JCL to be changed and formatted to meet specific site requirements. Because the JCL must be syntactically correct before using JCHG, we recommend that JEM or JSCAN be used first to validate the JCL.

#### *To change or reformat JCL with the JCHG Edit macro*

1. While in ISPF Edit on the JCL, type JCHG on the command line and press Enter.

JOB/SCAN displays the JCHG JOB/SCAN - CHANGE / REFORMAT JCL panel.

2. Specify which functions to execute and which Change or Reformat command member to use. Optionally, you may browse (B) or verify (V) commands before execution.
3. Inspect the member prior to saving it to be certain that Change modified the JCL as you wanted.
4. Save the changed or reformatted JCL by typing SAVE on the command line or to bypass saving, type CAN (cancel).
5. Press Enter.

If you saved your input, the modified JCL replaces the original input JCL in the library.

### Verifying Change Commands with JVER

The JVER Edit macro validates JCL Change and Reformatter commands for syntax errors only from within ISPF Edit, allowing immediate revision of the commands in response to flagged errors.

#### *To validate JCL Change or Reformatter commands*

1. While in ISPF Edit mode on the Change and Reformatter commands, type JVER on the command line and press Enter.

JVER validates the commands and redisplay them in Edit mode with appropriate error messages included as ISPF MSG lines.

2. Modify the commands as necessary.



Lines designated with MSG cannot be modified.

3. Repeats steps 1 and 2 until the Change commands are free of errors.
4. Press End to terminate the ISPF Edit session.

## Verifying Change Commands

Before running each command member, make sure you use JOB/SCAN's command verification facility. From the panels, use the `V` command, and from an Edit session use the `JVER` Edit macro. JOB/SCAN displays a message in the top right of the panel or Edit session if the commands are good. In the event that JOB/SCAN finds errors, it displays a report showing the line in error and describing the problem. JOB/SCAN will not run change commands that it cannot verify. In no case does correctness of the commands guarantee that the commands change JCL in the manner you want them to change.

From JOB/SCAN's panels use the `V` command; type the name of the command member and its library in the appropriate fields.

### Verifying Change commands from the Change Panel

```

----- JOB/SCAN - CHANGE / REFORMAT JCL -----
COMMAND ==> V

  F - Foreground Execution           E - Edit input JCL member
  B - Batch Execution                BR - Browse input JCL member
  V - Verify Chg/Ref Command Syntax  M - Modify/Create Chg/Ref Commands

Command Lib    ==> ASGUSR.CNTL
Member Name    ==> U3B012

Input Library  ==> ASGUSR.JCL
Input Type     ==> J             (J - JOB JCL, P - PROC, or L - Selection List)
Library Type   ==> PDS          (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> U3B012
Exclude Members ==>

Change JCL     ==> Y           (Y/N)
...

```

When editing a command member directly from its own library use the `JVER` Edit macro.



*Do not* use the `JVER` Edit macro from the Modify/Create panel.

If JOB/SCAN finds errors, it displays the member with messages relating to the first error it found. Quit this screen, correct the error, then rerun verification. The below figure contains a misspelling in the `ON EXEC` command.

### Viewing first error in a Change command member

```

ON EXE
DSS0185C - 'EXE' IS NOT A VALID 'ON' OPERAND
DSS0206W - SKIPPING TO 'END', THEN Syntax-CHECKING ONLY
IF DD=OUTPUT IS MISSING
ADD TIME=1440
EN
...

```

If there are more errors, JOB/SCAN re-displays the member with new messages. For example, in the below figure, END is misspelled "EN." Correct this error and rerun verification. Notice that JOB/SCAN inserts the message directly after the line in error.

#### Viewing subsequent error in a Change command member

```
...
ADD TIME=1440
EN
DSS0183C - EXPECTED AN OPERATOR; FOUND 'EN'
DSS0206W - SKIPPING TO 'END', THEN SYNTAX-CHECKING ONLY
ON DD
...
```



JOB/SCAN runs and verifies Change commands token by token. So, while it is permissible to write commands in this manner:

```
ON
DD
IF
UNIT=DASD
ADD
STMT
AFTER
UNIT=DASD
/*ADD THIS LINE
ENDADD
ADD
PARM=MYPARM
END
```

it yields commands that are less readable than these:

```
ON DD
IF UNIT=DASD
ADD STMT AFTER UNIT=DASD
/*ADD THIS LINE
ENDADD
ADD PARM=MYPARM
END
```

## Advanced Features

### MISSING Tests

Sometimes, in the course of maintaining JCL, it is necessary to test if a certain element is not present. Suppose your site standard requires that each DD have an explicit DISP parameter. You would not want to add the parameter if one was already there. MISSING tests help set the scope of an ON clause. Insert IF statements with MISSING tests after an ON clause; MISSING tests can also be combined with other IF tests.

The only ON operands under which a MISSING test can be included are: ON DD, ON EXEC, ON EXPGM, ON EXPROC, and ON JOB. Remember that adding or deleting JCL takes place in the scope established by the ON selector, conditional statement and prepositional phrases.

## MISSING Tests in ON EXEC Commands

If the active ON is an ON EXEC, ON EXPGM, or ON EXPROC, then the scope of the MISSING test is the current step. The only items you may test for MISSING status are DD and OUTPUT. Because a step might have several DD and OUTPUT statements, you may test for a specific DD or OUTPUT as missing. For example:

```
IF DD=SYSUDUMP IS MISSING
IF OUTPUT=OUT01 IS MISSING
```

## MISSING Tests in ON DD Commands

If the active ON is an ON DD, then the scope of the MISSING test is the current DD statement. You may test for MISSING status any DD statement parameter, for example, DISP, DSN, DCB and so on. You may also test for missing subparameters:

```
IF DCB:LRECL IS MISSING
```

Any given DD might have one DISP or DSN; so, you may not test for a specific value of a DD parameter or subparameter. Some examples might help. These tests are valid:

```
IF DISP IS MISSING
IF DCB IS MISSING
IF DCB:RECFM IS MISSING
```

But you cannot test for

```
IF DCB:RECFM=FB IS MISSING
```

because it doesn't make sense to test for a parameter or subparameter that can only occur once in a given statement. It is there or it is not and its value does not matter.

## MISSING tests in ON JOB Commands

If the active ON is an ON JOB, then the scope of the MISSING test is the current job and the items that you may test for MISSING status are shown in this table.

Statement	Example
EXEC	<pre>IF STEP=STEP015 IS MISSING IF EXPGM=IEBCOPY IS MISSING IF PGM=IEFBR14 IS MISSING IF EXPROC=RECOVER IS MISSING</pre>
PROC	<pre>IF PROC=MYPROC IS MISSING</pre>
Certain special (job-wide) DDs	<pre>IF DD=JOB CAT IS MISSING IF DD=JOB LIB IS MISSING</pre>
JCLLIB	<pre>IF JCLLIB IS MISSING</pre>



Statement	Example
JES2	<pre> IF JOBPARM IS MISSING IF MESSAGE IS MISSING IF NETACCT IS MISSING IF NOTIFY IS MISSING IF PRIORITY IS MISSING IF ROUTE IS MISSING IF SETUP IS MISSING IF SIGNOFF IS MISSING IF SIGNON IS MISSING IF XEQ IS MISSING IF XMIT IS MISSING </pre>
JES3	<pre> IF DATASET IS MISSING IF ENDDATASET IS MISSING IF ENDPROCESS IS MISSING IF MAIN IS MISSING IF NET IS MISSING IF NETACCT IS MISSING IF OPERATOR IS MISSING IF PAUSE IS MISSING IF PROCESS IS MISSING IF ROUTE IS MISSING IF SIGNOFF IS MISSING IF SIGNON IS MISSING </pre>



- Tests for missing items can be combined with each other and with other existing IF statements. For example:

```
IF PGM=PJGL0020 AND DD=OUTMAST IS MISSING
```

- In any MISSING test, the word NOT may be inserted to negate the condition. For example:

```
IF STEP=STEP015 IS NOT MISSING
```

- In all MISSING tests that allow the equal sign, the field immediately to the right of the equal sign may also be masked, using the standard JCL CHANGE wildcard characters (\* and %). (See [Wildcards](#) for more information.) For example:

```
IF EXPGM=IE%COPY IS MISSING
```

specifies that the subsequent action takes place only if no steps in the current job directly execute a program whose name consists of the characters *IE*, followed by any single character and the string *COPY*.



- It is intended, but not required, that the MISSING testing capability described in this section be used to determine the conditions under which a statement or line is to be added to the JCL. The MISSING tests may, however, be used to determine any *other* action except STRCHG.
- JES2, JES3, and JCLLIB tests for MISSING information cannot include an equal (=) sign.
- To determine if a parameter (not an entire statement) is missing, use:  

```
IF parameter-!=*
```

 instead of:  

```
IF parameter IS MISSING
```

### MISSING Tests Example 1

IF tests to set the scope of change commands might be further refined by the MISSING test. These commands:

---

```
ON EXEC
IF PGM=MYPROG AND DD=PCFDEL IS MISSING
ADD STMT AFTER MYPROG
//PCFDEL DD DSN=ASGUSR.PROD.CNTL
//      DCB=(LRECL=80,
//      RECFM=FA,
//      BLKSIZE=27920),
//      UNIT=3380,
//      DISP=(SHR,DELETE)
ENDADD
END
```

---

insert a DD immediately after the EXEC in this JCL:

---

```
//ASGUSRB JOB XXXX, 'JOBZ', NOTIFY=ASGUSR,
//      MSGCLASS=X, CLASS=F,
//      REGION=0M
//STEP1 EXEC PGM=MYPROG
//OUTPUT DD DSN=ASGUSR.OUTPUT
```

---

to produce this new JCL:

---

```
//ASGUSRB JOB XXXX, 'JOBZ', NOTIFY=ASGUSR,
//      MSGCLASS=X, CLASS=F,
//      REGION=0M
//STEP1 EXEC PGM=MYPROG
//PCFDEL DD DSN=ASGUSR.PROD.CNTL
//      DCB=(LRECL=80,
//      RECFM=FA,
//      BLKSIZE=27920),
//      UNIT=3380,
//      DISP=(SHR,DELETE)
//OUTPUT DD DSN=ASGUSR.OUTPUT
```

---

### MISSING Tests Example 2

This example shows what can happen when scope is not watched closely. At first glance it might appear that these commands are trying to add a UNIT parameter to a DD:

```
ON EXEC
IF DD=PCFDELX IS MISSING
ADD UNIT=3390
END
```

In fact, JOB/SCAN adds the UNIT parameter to the EXEC statement:

```
//STEP1 EXEC MYPROG,
//          UNIT=3390          <== IMPROPERLY PLACED UNIT PARM
//PCFDEL DD DSN=ASGPCF.USER1,
//          DCB=(BLKSIZE=27920)
```

Add a STRCHG command to alter the DCB line and change the ADD UNIT command to ADD STMT:

---

```
ON DD
STRCHG FROM 27920) TO 27920),
END
ON EXEC
IF DD=PCFDELX IS MISSING
ADD STMT AFTER 27920
//          UNIT=3390
ENDADD
END
```

---

JOB/SCAN does what you want:

```
//STEP1 EXEC PGM=MYPROG
//PCFDEL DD DSN=ASGPCF.USER1,
//          DCB=(BLKSIZE=27920),
//          UNIT=3390          <== PROPERLY PLACED UNIT PARM
```

### Annotating JCL

With the ANNOTATE verb, JOB/SCAN leaves comment lines in the JCL members it writes out to record the history of changes it made. (It does not annotate Reformatter modifications.) Optional user text provides additional detail. Later, you can use DEL ANNOTATE (form 4) to remove annotations. This section uses several examples to show common uses of the ANNOTATE verb.



- When processing at the JOB statement (MEMBER) level and JOB/SCAN makes no changes, JOB/SCAN annotates each member confirming that it was processed.
- JOB/SCAN does not annotate JCL members it only reformats; it must have changed the member in addition to the reformatting.

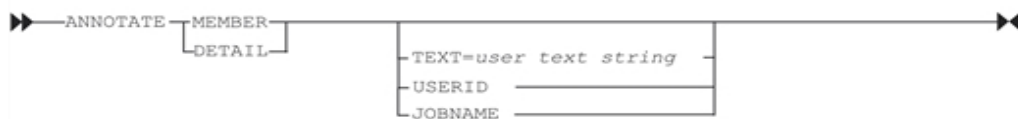
### Syntax

#### Form 1

```
▶▶ ANNOTATE DISABLE ◀◀
```

This form turns off all ANNOTATE verbs in a Change command member. This helps to avoid having to comment them out or remove them in order to use Change members and not annotate JCL. The remainder of the Change commands remain in force. Make ANNOTATE DISABLE the first Change command in the member.

#### Form 2



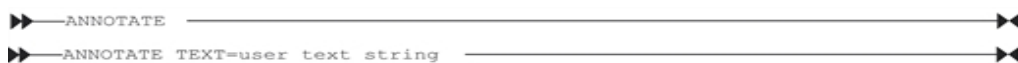
This form inserts annotations immediately after the JOB statement (MEMBER level) or after the individual statements changed (DETAIL level).

This form goes before the first ON command in a change command member. However, you may put an IF selector before Form 2.

Without user text, this form includes only date and time stamps. Enclose all user text strings in double quotes. User text strings in this form can be up to 47 characters long.

Use Form 2 when you want to add the name of the user ID under which the change is run or the JOB name to the annotation, or to add both.

### Form 3



This form inserts the date/time and optional user text immediately below the changed line (DETAIL level). Enclose text strings in double quotes. Text strings in this form can be up to 55 characters long. This form may go anywhere in the command member after an ON command and prior to the END delimiter. Its effects are restricted to the scope of the ON command in which it resides.

### Form 4



This form deletes all Change-inserted annotations at the MEMBER and DETAIL levels. Insert DEL commands after an ON command. Make no manual changes to Change-inserted annotations so that the DEL ANNOTATE command can recognize and remove annotations JOB/SCAN previously inserted.

### Example 1

Annotate at the MEMBER level; use Form 2. These commands:

```

ANNOTATE MEMBER
ON DD
  CHG DISP FROM SHR TO OLD
END

```

Change this JCL

---

```
//USRABCD JOB (ACCT),'ASG - J SMITH',
//      MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=ASGUSR
//STEP1 EXEC PGM=MYPGM
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=USR.SJJSPARM(DOCOPTM1),
//      DISP=SHR
//SYSUT2 DD SYSOUT=X
```

---

to this result:

---

```
//ASGUSR JOB (ACCT),'ASG - J SMITH',
//      MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=ASGUSR
//*PROCESSED - JOB/SCAN 2006/11/29 12:41:01
//STEP1 EXEC PGM=MYPGM
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=USR.SJJSPARM(DOCOPTM1),
//      DISP=OLD
//SYSUT2 DD SYSOUT=X
```

---

## Example 2

Annotate at the DETAIL level, using Form 2. These commands:

```
ANNOTATE DETAIL TEXT="CHANGED BY J.SMITH"
ON DD
CHG DISP FROM SHR TO OLD
END
```



The TEXT= option is used. Quotation marks are required only if the text contains spaces.

change this JCL

---

```
//ASGUSR JOB (ACCT),'ASG - J SMITH',
//      MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=ASGUSR
//*ROUTE PRINT ASG01
//STEP1 EXEC PGM=MYPGM
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=USR.SJJSPARM(DOCOPTM1),
//      DISP=SHR
//SYSUT2 DD SYSOUT=X
```

---

to:

---

```
//ASGUSR JOB (ACCT),'ASG - J SMITH',
//      MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=ASGUSR
//*ROUTE PRINT ASG01
//STEP1 EXEC PGM=MYPGM
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=ASG.#JSM.SJJSPARM(DOCOPTM1),
//      DISP=OLD
//* MODIFIED - JOB/SCAN 2006/11/29 12:41:01 CHANGED BY J.SMITH
//SYSUT2 DD SYSOUT=X
```

---

## Example 3

Annotate at both MEMBER and DETAIL levels, using Form 2.

### Change Commands

---

```
ANNOTATE MEMBER
```

---

```

ANNOTATE DETAIL
ON DD
  CHG DISP FROM SHR TO OLD
END

```

Change inserts these annotations: PROCESSED after the JOB statement; MODIFIED after each changed statement:

```

/** PROCESSED - JOB/SCAN 2006/11/29 10:14:16
/** MODIFIED - JOB/SCAN 2006/11/29 10:14:16

```



If ANNOTATE MEMBER and ANNOTATE DETAIL both are used with change rules affecting JOB statements, JOB/SCAN writes the DETAIL annotations after the JOB statement only.

#### Example 4

Delete ANNOTATE Comments, using Form 4.

You can delete annotations from the JCL at either the MEMBER level:

```

ON *
  DEL ANNOTATE-MEMBER
END

```

or DETAIL level:

```

ON *
  DEL ANNOTATE-DETAIL
END

```

or MEMBER and DETAIL levels:

```

ON *
  DEL ANNOTATE-MEMBER
  DEL ANNOTATE-DETAIL
END

```

#### Example 5

Turn off ANNOTATE Commands, using Form 1.

```

ANNOTATE DISABLE
ANNOTATE MEMBER
ANNOTATE DETAIL
ON EXEC
  ADD ROUTE=NXTLOCAL
END

```

This example does not insert any annotations at all. The DISABLE keyword overrides all ANNOTATE commands in the Change command member. The ADD command, however, is still in force; JOB/SCAN adds a ROUTE parameter to every EXEC statement.

#### Example 6

Conditional Annotation, using Form 3.

You can make annotations conditional by including ANNOTATE within an IF selector. Form 2 and Form 3 can be used in conditional annotations.

---

```

ON DD
  IF DISP=SHR
    CHG DISP FROM SHR TO OLD
    ANNOTATE TEXT="CHANGED BY J. DOE"
  END

```

---

JOB/SCAN, when it finds that DISP=SHR, inserts this comment after each changed DD statement:

```

//*MODIFIED BY JOB/SCAN 2006/11/29 14:15:13 CHANGED BY J. DOE

```



The ANNOTATE command need not come after the CHG command in the change string because, JOB/SCAN inserts an annotation only when it changes JCL, and, in this case, only when DISP=SHR.

### Example 7

Annotate with JOBNAME, using Form 2.

You can insert the name of the current JOB in the annotation; use the JOBNAME keyword. For this JCL:

```

//ASGABCD JOB (ACCT), 'ASG - J SMITH',
//      MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=ASGUSR
//*ROUTE PRINT ASG01

```

and these rules:

```

ANNOTATE MEMBER JOBNAME
ON DD
  CHG DISP FROM SHR TO OLD
END

```

JOB/SCAN inserts this comment after the JOB statement, and after each individual statement which was changed:

```

//*PROCESSED BY JOB/SCAN 2006/11/29 10:57:40 JOBNAME=ASGABCD

```

### Example 8

Annotate with USERID, using Form 2.

You can insert the name of the current user ID in the annotation; use the USERID keyword. With these commands:

```

ANNOTATE DETAIL USERID
ON DD
  CHG DISP FROM SHR TO OLD
END

```

JOB/SCAN inserts this comment for the user ASGUSR after the JOB statement, and after each individual statement changed:

```

//*MODIFIED BY JOB/SCAN 2006/11/29 10:57:40 USERID=ASGUSR

```

### Example 9

Annotate at DETAIL level with TEXT and &USERID Symbolic, using Form 2.

You can insert the name of the current user ID into user text; use the &USERID keyword.

```
ANNOTATE DETAIL TEXT="CHANGED BY &USERID"  
ON DD  
    CHG DISP FROM SHR TO OLD  
END
```

JOB/SCAN inserts this comment after each individual statement which was changed [if the &USERID symbolic is resolved as ASGDEF]:

```
//*MODIFIED BY JOB/SCAN 2006/11/29 10:57:40 CHANGED BY ASGUSR
```

### Example 10

Annotate at DETAIL level with TEXT and &JOBNAME Symbolic, using Form 2.

You can insert the name of the current JOB into user text; use the &JOBNAME keyword.

```
ANNOTATE DETAIL TEXT="CHANGED BY JOB &JOBNAME"  
ON DD  
    CHG DISP FROM SHR TO OLD  
END
```

JOB/SCAN inserts this comment after each individual statement it changed (because the &JOBNAME symbolic is resolved as ASGABCD):

```
//*MODIFIED BY JOB/SCAN 2006/11/29 10:57:40 CHANGED BY JOB ASGABCD
```



# JCL Reformatter

This topic describes the JOB/SCAN Reformatter facility (herein referred to as “Reformatter”).

- [Overview](#)
- [Setting Up Reformatter through Panels](#)
- [Creating Reformatter Command Members](#)
- [Writing Reformatter Commands](#)
- [Accessing Reformatter through Tables](#)
- [Running Reformatter in Foreground](#)
- [Running Reformatter with JREF Edit Macro](#)
- [Reformatting JCL with JCHG](#)
- [Running Reformatter in Batch Mode](#)
- [Reformatting Comments with the BC Table](#)
- [Removing Sequence Numbers from Columns 73-80](#)
- [Commenting a Reformat Member](#)

## Overview

Reformatter uses default and customized settings to automate the standardization of the formatting of JCL. Standardized format makes JCL easier to read, understand, and modify. The built-in defaults for Reformatter are stored in the B<sub>x</sub> Control Tables (where *x* is a letter from A through I). Administrators change the entries of these tables to adjust and refine the reformatting. Customized settings are specified in Reformatter commands. The combination of commands and tables (Reformatter commands override B<sub>x</sub> Control Tables) make a powerful, flexible facility.

ASG recommends the validation of the JCL before reformatting it to be sure the JCL syntax is correct. JOB/SCAN cannot properly reformat incorrect JCL.

### Reformatter Features

The JOB/SCAN Reformatter provides the capability to:

- Place the operation field in a specific column.
- Specify the beginning column of the first parameter for each statement.
- Specify the number of parameters per line.
- Specify the order in which parameters should appear.
- Re-sequence or eliminate sequence numbers in columns 73 through 80.
- Restructure or eliminate comment fields.
- Realign EXEC and DD parameters so that all JCL is formatted uniformly.
- Restructure PROC and PEND JCL statements.
- Restructure SET, JCLLIB, INCLUDE, EXPORT, and OUTPUT JCL statements.

## Using Reformatter

Using Reformatter involves:

- Defining the Reformatter settings through commands and tables.
- Ensuring the JCL is syntactically correct.
- Invoking Reformatter.
- Reviewing the reformatted JCL.

Reformatter can reformat members of:

- PDS libraries
- PANVALET libraries
- LIBRARIAN libraries
- Sequential datasets

## Reformatting Execution Options

Execute Reformatter in one of these ways:

- From the ASG-JOB/SCAN Primary Menu panel.
- With either the JREF or JCHG edit macros.
- In a batch job. See [Validating JCL in Batch](#).

Reformatter reads the members specified, applies the defined Reformatter options, and creates reformatted output members with the same member names.



Because newly reformatted members can overwrite members in the input library, ASG strongly recommends using a different library for output.

Reformatting options can be specified as commands or through Control Table entries in an SJSPARM library member (typically JSOPT02).

## Reformatter Processing Rules



Thoroughly test all Reformatter commands and tables before committing to production, especially when reformatting comment fields, to ensure that all aspects of JCL are handled the in the manner you intend.

These processing rules apply to all JCL reformatting:

- When Reformatter encounters duplicate Control Table entries, it uses the first entry.
- When Reformatter encounters duplicate commands, it uses the last command.
- Reformatter commands override Control Table entries.

- Reformatter does not have default reformatting for non-standard parameters or symbolic parameters. When it encounters these parameters without reformatting specified for them in the Control Tables, its default placement for these parameters is the last record of the JCL statement after all other specified statement reformatting.
- Reformatter, either commands or tables, works only from the Change/Reformatter panel, but does not work from the JCL Validation panel.



To support CONTROL-M Auto Edit statements (%%IF, %%ELSE, and %%ENDIF), specify the Program Product Interface (PPI=CONTROL-M) Run Option. For more information about the Program Product Interface, see [Program Product Interface \(PPI\)](#).

## Setting Up Reformatter through Panels

This section describes how to setup the JOB/SCAN JCL Reformatter using the Change/Reformat JCL panels.

Begin with the first Change/Reformat JCL Panel (J02PCRI). See figure - [First Change/Reformat JCL Panel](#).

### **To access the Reformatter facility panels and setup Change/Reformat JCL**

1. Select the Change/Reformat JCL option from the ASG-JOB/SCAN Primary Menu.  
JOB/SCAN displays the first Change/Reformat JCL panel.
2. Choose the execution mode by typing it on the command line. This table describes the modes.

Option	Meaning
F	Execute functions in Foreground.
B	Execute functions in Batch.
V	Verify the reformatter commands.
E	Edit input JCL members in ISPF Edit mode.
BR	Browse input JCL members in ISPF browse mode.
M	Create new or Modify existing reformatter command member in ISPF Edit mode.

3. Enter the locations of the Reformatter change commands and JCL input libraries. This table describes the fields.

Panel Field	Meaning and Input
COMMAND LIB	Full dataset name for the change command member.

Panel Field	Meaning and Input
MEMBER NAME	Name of the member containing the change commands.
INPUT LIBRARY	High level qualifier for input JCL.
INPUT TYPE	Type of JCL: J – for job JCL P – for procedure JCL L – for selection list.
LIBRARY TYPE	Type of library: P – for PDS PAN – for PANVALET LIB – for LIBRARIAN LIB-3 – for LIBRARIAN 3 SEQ – for sequential datasets.
SELECT MEMBERS	JCL input members to be processed. (To change an entire library, type ALL in the Select Members field.)
EXCLUDE MEMBERS	JCL input members to excluded from members selected.

- Choose whether to change the JCL, reformat the JCL, or validate the JCL, or any combination. For Reformatter, set Reformat JCL to Y; set the others to N.

If more than one of the fields is set to Y, the execution sequence is:

- Change the JCL.
  - Reformat the JCL.
  - Validate the changed or reformatted JCL.
- Optionally, enter any report header texts.

**First Change/Reformat JCL Panel**

```

J02PCRI ----- JOB/SCAN - CHANGE / REFORMAT JCL -----
COMMAND ==>

      F - Foreground Execution          E - Edit input JCL member
      B - Batch Execution               BR - Browse input JCL member
      V - Verify Chg./Ref Command Syntax M - Modify/Create Chg/Ref Commands

Command Lib    ==> TEST.CNTL
Member Name   ==> GOFORIT

Input Library  ==> 'DSSI.#COM2.LIST'
Input Type    ==> L          (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ==> PDS       (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> LIST1A
Exclude Members ==>

Change JCL    ==> N   (Y/N)
Reformat JCL  ==> Y   (Y/N)
Validate JCL  ==> N   (Y/N)

Report Heading ==>

```

- After you have completed the entries for the first Change/Reformat JCL panel, press Enter to proceed to the second Change/Reformat JCL panel (J02PCRO).

See the table in step [step](#) for descriptions of the fields of this panel.

**Second Change/Reformat JCL Panel**

```

J02PCRO ----- JOB/SCAN - CHANGE / REFORMAT JCL -----
COMMAND ==>

Write Changed JCL? ==> Y          (Yes/No/ChgOnly)

Output Lib      ==> WORK.CNTL
Library type    ==> PDS          (PDS, PAN, LIB, LIB-3, or SEQ)

JCL Change Report:
Display         ==> O (OutJCL/Audit/ChgOnly/None)
Hardcopy       ==> N (OutJCL/Audit/ChgOnly/None)

Processing Summary:
Display        ==> Y (Y/N)
Hardcopy       ==> N (Y/N)

Press <ENTER> to continue.
Press <END> to cancel.

```

- Specify whether JCL Reformatter writes the reformatted JCL. If so, specify the output library and its type.
- Specify the reports for JOB/SCAN to produce, and whether to print or display them.

**Panel Field****Meaning and Input**

OUTPUT LIBRARY

The high level qualifier where JOBSKAN writes the output members.

Panel Field	Meaning and Input
LIBRARY TYPE	<p>Type of library:</p> <p>P – for PDS</p> <p>PAN – for PANVALET</p> <p>LIB – for LIBRARIAN</p> <p>LIB-3 – for LIBRARIAN 3</p> <p>SEQ – for sequential datasets</p>
JCL CHANGE REPORT: DISPLAY HARDCOPY	<p>Type of report to display, print, or both:</p> <ul style="list-style-type: none"> <li>• O (OutJCL) – Member showing changed statements.</li> <li>• A (Audit) – Member showing statements before and after change.</li> <li>• C (Change only) – Listing showing only changed statements before and after.</li> <li>• N (None) – Neither hardcopy nor display.</li> </ul>
PROCESSING SUMMARY: DISPLAY HARDCOPY	Disposition of the Summary Report: Display, print, or both.

When Reformatter foreground execution is complete, JOB/SCAN displays the reformatted JCL or the summary report if one was specified. The JCL Output Report displays a \$ symbol over the first / (slash) of all changed statements. The Audit Report and the Processing Summary Report display an O (old), N (new), A (added), or D (deleted) to the left of each changed or reformatted statement.

When Reformatter background execution is complete, view the held output.



If the value for the Validate JCL and Change JCL fields on the first Reformat panel are N, Reformatter begins processing the JCL immediately. If the value for the Validate JCL field is Y, JOB/SCAN displays the ASG-JOB/SCAN - Validate Changed/Reformatted JCL panel. This panel is an abridged version of the ASG-JOBSCAN - JCL Validation panel.



### Validate Changed Reformatted JCL panel.

```

----- ASG-JOB/SCAN - VALIDATE CHANGED/REFORMATTED JCL -----
COMMAND ===>

      blank - Validate JCL
          P - Processing Options
          S - Submit After Validation

First Proclib      ===>
  Proclib Type    ===>                (PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
  Display        ===> N (Yes/No/ErrorsOnly)  Browse/View ===> B (B/V)
  Hardcopy       ===> Y (Yes/No/ErrorsOnly/SpaceOnly/JobErrors)

Options in Effect Listing  ===> Y (Y/N)          SJL PDSOUT ===> Y (Y/N)

Selection Exit  ===>                Parm  ===>

Standards Name  ===>                Parm  ===>

Press <ENTER> to continue.  Press <END> to cancel.

```

Make changes to the fields in the ASG-JOB/SCAN - Validate Changed/Reformatted JCL panel then press Enter to continue processing.

## Creating Reformatter Command Members

Create or modify Reformatter commands with the command option M on the first ASG-JOB/SCAN - Change/Reformat JCL panel. It starts an ISPF Edit session. After a member is created and saved in the command library, it can be modified at any time using the same procedure.

### To create or modify Reformatter commands

1. On the first JOB/SCAN - Change/Reformat JCL panel, type M on the command line.
2. In the Command Lib field, type the name of the command library.



The command library must have a logical record length (LRECL) of 80.

3. In the Member Name field, type the name of the member containing the Reformatter commands. Leave the field blank to display a directory of members.
  4. Press Enter to process.
    - If a Member Name was specified, an ISPF Edit panel appears. If the member does not exist, the Edit session starts on a blank member.
- Or**
- If the Member Name field was left blank, a directory of member names appears. Select the desired member.

5. Using standard ISPF Edit functions, type the Reformatter commands.
6. Press End when the command group is completed to save the member and exit.
7. Verify the commands before using them with the V command option. See [Verifying Reformatter Commands](#).

## Writing Reformatter Commands

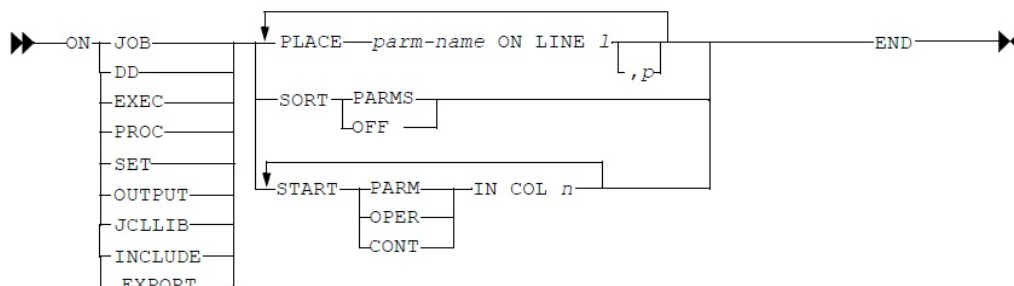
Reformatter commands may reside in a JCL Change command member or in a separate Reformatter member. Reformatter commands override the Reformatter Control Tables (BA through BJ), referred to herein as the B<sub>x</sub> Control Tables.

Reformatter defaults are stored on the Reformatter Control Table entries. All reformatting capabilities offered through Reformatter commands are available in the Reformatter Control Tables. The JOB/SCAN Product Administrator controls access to Reformatter commands and Control Tables.

### General Syntax

#### Reformatting JCL Statements

JCL Statements can be reformatted using this Reformatter command:



where *l* is the line number (from 0 to 99), *n* is the column number (from 4 to 72), *p* is the parameter's position (1-99) on line, and *parm-name* is a keyword.

Here are some examples of Reformatter commands:

- Put DISP as the first parameter on line 1 and DSN on line 2 for every DD in the job:

```
ON DD
  PLACE DISP ON LINE 1,1
  PLACE DSN  ON LINE 2
END
```

- Start all continuation statements in column 5 and all operators in column 12 for every PROC in the job:

```
ON PROC
  START CONT IN COL 5
  START OPER IN COL 12
END
```

- Start all parameters in column 12 for every OUTPUT statement in the job:

```
ON OUTPUT
  START PARM IN COL 12
END
```

#### Reformatting JCL Comment Statements

JCL comment statements can be reformatted using this Reformatter command:





- Renumber JCL statements starting with line number 100 and incrementing by 10:

```
ON JCL RENUM FROM 100 BY 10
```

- Remove numbering from JCL statements:

```
ON JCL UNNUM
```

### Reformatter Command Example

Examine the before and after examples to see the effects of the Reformatter commands.

Reformatter Commands	JCL Before Reformatting	JCL After Reformatting
ON DD	//SYSUT2 DD DSN=&&TEMP,	//SYSUT2 DD DISP=(,PASS),
PLACE UNIT ON LINE 5	//	DCB=BLKSIZE=6160,
PLACE DISP ON LINE 1	//	DISP=(,PASS),
PLACE DCB ON LINE 2	//	UNIT=SYSDA,
PLACE SPACE ON LINE 3	//	SPACE=(TRK,(5,1)),
PLACE DSN ON LINE 4	//	DSN=&&TEMP,
END	//	DCB=BLKSIZE=6160
		//
		UNIT=SYSDA

### Usage Notes for Reformatter Commands

- Because Reformatter does not currently support IF logic, Reformatter commands are not allowed after IF selectors in JCL Change members.
- Reformatter commands override Control Table entries. Use Control Tables to specify site default and use Reformatter commands for special-case overrides.
- Control Table BC offers only one field to specify the start column for operators for the several types of JCL statements; use Reformatter commands to specify different start columns for each statement type.
- Do not attempt to reformat the same object in incompatible ways. For example, only one `PLACE COMMENT-FIELD` should be coded. If more than one is coded, the last one specified is used. Likewise, `START COMMENT-FIELD IN COL n` does not work with `PLACE COMMENT-FIELD AS SENTENCE` or with `PLACE COMMENT-FIELD USING TEMPLATE n`. However, `START COMMENT-FIELD IN COL n` works fine with `PLACE COMMENT-FIELD WITH xxxx PARM`.
- SORT alphabetically sorts (descending) un-reformatted parameters and places them ahead of otherwise reformatted parameters.
- Reformatter places all the unspecified parameters ahead of otherwise reformatted parameters.
- Although it is permitted to use numbers as big as 10,000,000-1 in renumbering, in practice, it is of no practical merit.
- When it encounters repeated commands, JOB/SCAN uses the last one.
- JOB/SCAN does not reformat comment statements or fields on un-reformatted statements.
- The JCL Change command `DEL COMMENT-FIELD` command supersedes the Control Table BC entry when seeking to delete comment fields.

## Accessing Reformatter through Tables

Reformatting options can be specified through Control Table entries in a PARMLIB member. They may also be specified as commands, as shown in the table. See [Writing Reformatter Commands](#) for more details. Also, see [Control Tables](#) for complete details about Reformatter Tables.

### Entries in Control Table Bx

This table lists Reformatter Bx Control Tables. See [Control Tables](#) for complete information about configuring Control Tables.

#### Bx Control Table

Control Table	Description
BA	Defines the relative line and position of specific DD statement parameters.
BB	Defines the relative line and position of specific EXEC statement parameters.
	For all statement types, entries in this control table can: <ul style="list-style-type: none"> <li>identify the starting position for:               <ul style="list-style-type: none"> <li>the operation field</li> <li>the first parameter on the operation field line</li> </ul> </li> </ul>
BC	continuation lines <ul style="list-style-type: none"> <li>alter or remove the line sequence number in positions 73 through 80 of JCL statements processed</li> <li>reorganize comments and comment fields</li> <li>remove comment fields.</li> </ul>
BD	Defines the relative line and position of specific PROC parameters.
BE	Defines the relative line and position of specific JOB statement parameters.
BF	Defines the relative line and position of specific OUTPUT statement parameters.
BG	Defines the relative line and position of specific SET statement parameters.
BH	Defines the reformatting option for JCLLIB statements.
BI	Defines the reformatting option for INCLUDE statements.
BJ	Defines the reformatting option for EXPORT statements.

The effect of settings and entries in the BC table upon particular statement types depends on the presence of entries in other Bx Control Tables.

For example, coding a BC table to reformat JOB statement will have no effect without a corresponding BE table entry. To prevent reformatting of a JCL statement type, remove the appropriate Control Table entries from the SJJSPARM library. For example, remove all BA tables to turn off DD statement reformatting. See [Control Tables](#) for details about Control Table reformatting.

## Examples of Reformatting Using Control Tables

Basic reformatting examples using Control Tables:

- Placing DISP on line 1 and DSN on line 2 for every DD:

```
....+....1....+....2....+....3....+....4....+....5....+
      BA01,DISP
      BA02,DSN
```

- Starting all continuation statements in column 5 and all operators in column 11 for every DD within the Job:

```
....+....1....+....2....+....3....+....4....+....5....+
      BA
      BC                11    05
```

- Placing comment statements with the last parm:

```
....+....1....+....2....+....3....+....4....+....5....+
      BC                2      L
```

- Starting all comment fields in column 50:

```
....+....1....+....2....+....3....+....4....+....5....+
      BC                2      S50
```

- Starting all JOB continuation lines in column 15:

```
....+....1....+....2....+....3....+....4....+....5....+
      BC                15
      BE
```

The BE table turns on reformatting of the JOB statement.

- Renumbering JCL statements starting with line number 100, incrementing by 10:

```
....+....1....+....2....+....3....+....4....+....5....+
      BC00100,00010
```

- Removing the numbering from JCL statements:

```
....+....1....+....2....+....3....+....4....+....5....+
      BC00000
```

Placing parameters on the same line and specifying a relative position. For example, to reformat the DSN and DISP parameters to the same line with DSN always appearing first, code a command then the relative position after the line number:

```
....+....1....+....2....+....3....+....4....+....5....+
      BA01,01DSN
      BA01,02DISP
```

## Control Table and Reformatter Command Equivalents

This table shows samples of Reformatter commands that perform the same functions as the equivalent B<sub>x</sub> Control Table entries.

Control Table	Equivalent Command Examples
BA	ON DD PLACE DSN ON LINE 1 PLACE DISP ON LINE 1 PLACE VOL ON LINE 2

Control Table	Equivalent Command Examples
BB	<pre> PLACE UNIT ON LINE 99 END  ON EXEC PLACE REGION ON LINE 1,1 PLACE COND ON LINE 1,2 PLACE PARM ON LINE 2 SORT PARMS END  ON JCL RENUM FROM 100 BY 10 PLACE COMMENT-LINE ON BOTTOM LINE PLACE COMMENT-FIELD WITH FIRST PARM START COMMENT-FIELD IN COL 40 END ON DD START OPER IN COL 12 START PARM IN COL 18 START CONT IN COL 16 END </pre>
BC	<pre> ON EXEC START OPER IN COL 12 START PARM IN COL 18 START CONT IN COL 12 END ON PROC START OPER IN COL 11 START PARM IN COL 16 START CONT IN COL 11 END ON JOB START OPER IN COL 11 START PARM IN COL 16 START CONT IN COL 11 DEL COMMENT-FIELD END </pre>
BD	<pre> ON PROC SORT PARMS PLACE DATEPARM ON LINE 1 PLACE TAPEVOL ON LINE 6 END </pre>
BE	<pre> ON JOB PLACE CLASS ON LINE 2,1 PLACE MSGCLASS ON LINE 2,2 PLACE NOTIFY ON LINE 3 SORT PARMS END </pre>



- If multiple parameters are defined to Reformatter as being on the same line and position, they will be placed on that line in the order in which they occur in the input JCL.
- The sequence of parameters can be specified for two or more parameters reformatted to the same line. In Control Table BA, code a comma after the relative line number and specify a relative position. For example, to reformat the DSN and DISP parameters to the same line with DSN always appearing first, code:

```

.....+.....1.....+.....2.....+.....3.....+.....
BA01,01DSN
BA01,02DISP

```



- To prevent reformatting of DD, EXEC, PROC, JOB, or OUTPUT statements, remove Control Table Bx entries from the SJJSPARM library members.
- If there is insufficient space on the specified line for assigning multiple parameters, the remaining parameters for that line overflow to subsequent lines.

## Running Reformatter in Foreground

This section describes how to run JOB/SCAN JCL Reformatter interactively from ISPF panels.

### Modes of Operation

Reformatter runs in foreground in any of these ways:

- Verify mode.
- Simulation (test) mode.
- Write mode.

If you proceed with reformatting tasks in the order listed here, you will avoid common errors. After creating the commands, verify them; then, test them to make sure they do what you want them to do; then put them into production and write the reformatted members to a new library; then, lastly validate them with JOB/SCAN.

### Verifying Reformatter Commands

There are two methods of verifying the Reformatter command syntax: using the V (verify) option from the command line on the first Change/Reformat JCL panel, or using the JOB/SCAN Edit macro JVER while editing the command member in ISPF Edit.

Because JCL Reformatter commands do not execute if the command syntax is incorrect, always verify any new Reformatter commands before applying them to a JCL member, to be sure they are free of JCL Reformatter syntax errors.

### Verifying Commands from the Change/Reformat JCL Panels

#### *To verify the syntax of Reformatter commands*

1. On the first Change/Reformat JCL panel, type `v` on the command line. Be sure the correct command library and member name are in the appropriate fields, and press Enter.

If errors are found, a listing of the Reformatter commands appears with the errors flagged.

2. Examine the errors detected by the verification step.
3. Press End to return to the first Change/Reformat JCL panel.
4. To correct any errors, type `M` on the command line and press Enter.
5. Make corrections and press End.
6. Repeat steps 1 through 5 until no more errors are found.

## Verifying Commands with the JVER Edit Macro

The JVER Edit macro validates JCL Change and Reformatter commands from within ISPF Edit, allowing immediate revision of the commands in response to flagged errors.

### To validate JCL Change or Reformatter commands

1. While in ISPF Edit mode on the Change and Reformatter commands, type `JVER` on the command line and press Enter.

JVER validates the commands and re-displays them in Edit mode, with appropriate error messages included as ISPF MSG lines.

2. Modify the commands as necessary.

 Lines designated with MSG cannot be modified.

3. Repeats steps 1 and 2 until the Reformatter commands are free of errors.
4. Press End to terminate the ISPF Edit session.

### Running Reformatter in Batch Mode from Panels

To start JCL Reformatter from the panels to run in batch mode instead of in the foreground, follow the procedure given for foreground execution, but change the execution option from F (foreground) to B (batch) on the command line of the first Change/Reformat JCL panel.


On the bottom of the first Change/Reformat JCL panel, the three fields: Change JCL, Reformat JCL, and Validate JCL must each have a value, either Y for yes, or N for no, and at least one must be Y. If all three are set to Y, the execution sequence is: change the JCL first, reformat it second, and validate the changed and reformatted JCL last.

The selection characters \* and % are not allowed in the Member Name field.

Reformatter commands cannot be verified in batch. Instead, create and test commands using the Change/Reformat JCL panels in ISPF; or use the JVER and JCHG Edit macros to create and test the commands. Then execute the Reformatter function in batch.

### Simulating Reformatter

JCL Change simulation mode allows testing of JCL Change commands before changing a JCL member. The resulting output listing shows the complete JCL member as it would be when changed. The changed statements are flagged with a JCL Change Replacement character, typically a \$ (dollar sign) in column 1.

-  The Product Administrator may change the default JCL Change Replacement Character (\$) to a different character.
- See [Setting Up Reformatter through Panels](#) for more information about the Change/Reformat JCL panels.

### To simulate reformatting

1. From the ASG-JOB/SCAN Primary Menu, select the Change/Reformat JCL option.  
The first Change/Reformat JCL panel appears.

2. Type **F** on the command line.
3. Type the name of the command library and member in the Command Lib and Member Name fields. To obtain a directory of the command library, leave the Member Name field blank.
4. Fill in the Input Library, Input Type, and Library Type fields. Use the Select Members and Exclude Members fields to specify the input JCL members, or leave them blank to display the directory. See [Selecting Members](#).
5. Type **Y** in the Reformat JCL field.
6. Press Enter to proceed to the second JOB/SCAN - Change/Reformat JCL panel.
7. On the second JOB/SCAN - Change/Reformat JCL panel, type **N** in the Write Changed JCL? field. This makes the run a simulation.
8. Type **O** (for output JCL report) in the JCL Change Report Display field, and **N** in the JCL Change Report Hard-copy field.
9. Press Enter. When the simulation is complete, JOB/SCAN displays an output listing.



You can also simulate reformatting from the panel using the batch (**B**) option in Step 2. Type **N** in the JCL Change Report Display field and **O** (for output JCL report) in the JCL Change Report Hard-copy field. Then, review results from the held output.

The results of a batch run on the input JCL show in figure - [Reformatter Simulation Mode - JCL Before Simulation \(DD name JLCHLI\)](#) are shown in figure - [Reformatter Simulation Mode - JCL After Simulation \(DD name JLCHLO\)](#). Both figures show only a portion of the full report.

#### Reformatter Simulation Mode - JCL Before Simulation (DD name JLCHLI)

```

JCL REFORMAT INPUTS          ASG TECHNOLOGIES GROUP, INC.          PAGE      2
-----
|                               AS OF: 05/10/2005          |
| INPUT   ==> ASGUSR.SJJSDEMO (FORMJCL)                   |
|                               07:19:04                  |
-----
//FORMAT JOB (A,P,GL), 'MESS', CLASS=P, MSGCLASS=H
//* -----
//* (FORMJCL) THIS MEMBER IS USED TO DEMONSTRATE THE REFORMATTER
//*          FUNCTION OF JOB/SCAN.
//* -----
//PS010 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=TAPE9, LABEL=(,NL), DISP=OLD, UNIT=TAPE9,
// VOL=SER=NEEDED
//SYSUT2 DD DSN=&&TEMP, DISP=(,PASS), UNIT=SYSDA, SPACE=(TRK,(5,1)),
// DCB=BLKSIZE=6160
//SYSUDUMP DD SYSOUT=A
//PS020 EXEC PGM=SORT, REGION=512K
//STEPLIB DD DSN=DSSI.WORK.LOAD, DISP=SHR
//SORTLIB DD DSN=SYS1.SORTLIB, DISP=SHR
//SORTPRD DD SYSOUT=(J,,9), COPIES=2
//SYSPRINT DD SYSOUT=A
//SYSIN DD DSN=DSSI.GLPROD.CNTL(G201001S), DISP=SHR
//SORTIN DD DSN=*.PS010.SYSUT2, DISP=(OLD,DELETE)
...

```



## Reformatter Simulation Mode - JCL After Simulation (DD name JLCHLO)

```

JCL REFORMAT OUTPUT LISTING      ASG TECHNOLOGIES GROUP, INC.                PAGE      1
-----
|OUTPUT   ==>                               AS OF: 05/10/2005 |
|INPUT    ==> ASGUSR.SJJSDEMO (FORMJCL)      07:19:04   |
|COMMANDS ==>                               |
-----
//FORMAT   JOB (A,P,GL), 'MESS', CLASS=P, MSGCLASS=H
//* -----
//* (FORMJCL) THIS MEMBER IS USED TO DEMONSTRATE THE REFORMATTER
//*          FUNCTION OF JOB/SCAN.
//* -----
//PS010    EXEC   PGM=IEBGENER
//SYSPRINT DD   SYSOUT=A
//SYSUT1   DD   DSN=TAPE9,
//          DISP=OLD,
//          UNIT=TAPE9,
//          LABEL=(,NL),
//          VOL=SER=NEEDED
//SYSUT2   DD   DSN=&&TEMP,
//          DISP=(,PASS),
//          UNIT=SYSDA,
//          SPACE=(TRK,(5,1)),
//          DCB=BLKSIZE=6160
//SYSUDUMP DD   SYSOUT=A
//PS020    EXEC   PGM=SORT,
//          REGION=512K
//STEPLIB  DD   DSN=DSSI.WORK.LOAD,
//          DISP=SHR
//SORTLIB  DD   DSN=SYS1.SORTLIB,
//          DISP=SHR
//SORTPRT  DD   SYSOUT=(J,,9),
//          COPIES=2
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   DSN=DSSI.GLPROD.CNTL(G201001S),
//          DISP=SHR
//SORTIN   DD   DSN=*.PS010.SYSUT2,
//          DISP=(OLD,DELETE)
...

```

## Writing Output Members

This section describes how to run JOB/SCAN JCL Reformatter from the ISPF panels, creating reformatted JCL members. See [Setting Up Reformatter through Panels](#) for more information about the Change/Reformat JCL panels.

When you are satisfied (through simulation) that the JCL Reformatter commands and tables produce the desired results, you can now create updated JCL members.




ASG *strongly* recommends *not* overwriting the input JCL with the reformatted output JCL. Instead, write the output JCL to a different library and validate from this library.

## To create an updated JCL member

1. From the ASG-JOB/SCAN Primary Menu, select the Change/Reformat JCL option. The first Change/Reformat JCL panel appears.

2. Type **F** on the command line.
3. Type the name of the command library and member in the Command Lib and Member Name fields.

 To obtain a directory of the command library, leave the Member Name field blank.

4. Fill in the Input Library, Input Type, and Library Type fields. Use the Select Members and Exclude Members fields to specify the input JCL members, or leave them blank to display the directory.

See [Selecting Members](#).

5. Type **Y** in the Reformat JCL field.
6. Press Enter. JOB/SCAN displays the second Change/Reformat JCL panel.
7. On the second Change/Reformat JCL panel, type **Y** (Yes) or **C** (ChgOnly) in the Write Changed JCL? field. This causes JOB/SCAN to write reformatted JCL to an output member.
8. In the Output Lib field, type the name of an existing library to receive the changed JCL.
9. Specify the type of output library in the Library type field. Valid values are SEQ, PDS, PAN, LIB, and LIB-3.
10. Specify Processing Summary output options. In Foreground, type **Y** in the Display field. Select **B** to display in Browse mode or **V** to display in View mode. In Batch (background), type **Y** in the Hardcopy field.

#### Writing Changed JCL to an Output Member

```

----- JOB/SCAN - CHANGE / REFORMAT JCL -----
COMMAND ==>

Write Changed JCL?  ==> Y          (Yes/No/ChgOnly)

Output Lib         ==> WORK.CNTL
Library type       ==> PDS          (PDS, PAN, LIB, LIB-3, or SEQ)

JCL Change Report:
  Display          ==> O (OutJCL/Audit/ChgOnly/None)
  Hardcopy         ==> N (OutJCL/Audit/ChgOnly/None)

Processing Summary:
  Display          ==> Y (Y/N)
  Hardcopy         ==> N (Y/N)

Press <ENTER> to continue.
Press <END> to cancel.

```

11. Press Enter.

## Running Reformatter with JREF Edit Macro

The JREF Edit macro invokes JCL Reformatter from within ISPF Edit, allowing JCL to be formatted to meet specific site requirements. Because the JCL must be syntactically correct before using JREF, ASG recommends that JEM or JSCAN be used first to validate the JCL.

### To reformat JCL with the JREF Edit macro

1. While in ISPF Edit on the JCL, type `JREF` on the command line and press Enter. JOB/SCAN automatically reformats and displays the JCL currently being edited.
2. To save the reformatted JCL, type `SAVE` on the command line and press Enter.

The reformatted JCL replaces the original input JCL in the library.

3. To bypass saving the reformatted JCL, type `CAN` (for cancel) on the command line, and press Enter.

### Example of JCL After Formatting

```
EDIT ---- DSSI.PROC.CNTL(PJGL0072) - 01,06 -----CIKYNBS 001 072
COMMAND ==>                                SCROLL ==> CSR
***** ***** TOP OF DATA *****
000001 //PJGL0072 JOB (A,P,PR),'DSSI-PRINT',
000002 //          NOTIFY=DSSI,CLASS=A,
000003 //          TIME=1440
000004 //STEP010 EXEC PGM=IEBGENER
000005 //SYSUT1  DD DSN=ASGUSR.WORK.CNTL,DISP=OLD
000007 //SYSUT2  DD DSN=DSSI.DEMODSN2,DISP=(,CATLG,DELETE),
000009 //          UNIT=SYSDA,SPACE=(TRK,(5,1,RLSE)),
000011 //          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
000012 //SYSPRINT DD          SYSOUT=A
000013 //*
```

### Example of JCL After Formatting

```
EDIT ---- DSSI.PROD.CNTL(PJGL0072) - 01.07 ----- COLUMNS 001 072
COMMAND ==>                                SCROLL ==> CSR
***** ***** TOP OF DATA *****
000001 //PJGL0072 JOB (A,P,PR),'DSSI-PRINT',
000002 //          NOTIFY=DSSI,CLASS=A,
000003 //          TIME=1440
000004 //STEP010 EXEC PGM=IEBGENER
000005 //SYSUT1  DD DSN=ASGUSR.WORK.CNTL,
000006 //          DISP=OLD
000007 //SYSUT2  DD DSN=DSSI.DEMODSN2,
000008 //          DISP=(,CATLG,DELETE),
000009 //          UNIT=SYSDA,
000010 //          SPACE=(TRK,(5,1,RLSE)),
000011 //          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
000012 //SYSPRINT DD          SYSOUT=A
000013 //*
```

## Reformatting JCL with JCHG

The JCHG Edit macro invokes Reformatter from within ISPF Edit, allowing JCL to be formatted to meet specific site requirements. Because the JCL must be syntactically correct before using JCHG, ASG recommends validating the JCL with JEM or JSCAN before using JCHG.

## Using JCHG

### To reformat JCL with JCHG while in ISPF Edit on the JCL

1. Type `JCHG` on the command line and press Enter.

The JOB/SCAN - CHANGE / REFORMAT JCL panel is displayed. It allows the user to specify which functions to execute and which Reformat commands to use.

2. Optionally browse or verify the commands before execution.

Reformatter does not require a command library member. If the Reformatter is only executed, and a member is not specified, the Reformatter uses Bx table entries to reformat the JCL. If a member is specified, the Reformat commands override the Bx table entries.

3. JOB/SCAN reformats and displays the JCL in the Edit session.
4. To save the reformatted JCL, type `SAVE` on the command line and press Enter. The modified JCL replaces the original input JCL in the library.



If you are not interested in saving the modified JCL, type `CAN` (Cancel) on the command line; press Enter.

### JOB/SCAN - Change / Reformat JCL Panel

```

----- JOB/SCAN - CHANGE / REFORMAT JCL -----
OPTION ==>

    blank - Change/Reformat the JCL
         B - Browse Change/Reformat Commands
         V - Verify Change/Reformat Command Syntax

Command Library ==> CNTL
  Member Name ==> U3B012M   (Blank for member selection list)

Change JCL    ==> Y   (Y/N)
Reformat JCL  ==> N   (Y/N)

Press ENTER to continue, or END to terminate.

```

## Running Reformatter in Batch Mode

If Reformatter is run against a particularly large library, run it in batch from the panels, or submit a batch job using the SJJSSAMP library member JSMVS.

The SJJSSAMP library contains these members for running JCL Reformatter in batch:

- JSMVS — executable JCL to run JOB/SCAN in batch.
- JS6PROC — the PROC invoked by JSMVS.

## What the JSMVS Member Does

The SJSSAMP library member JSMVS contains the executable JCL to run JCL Reformatter in batch mode. JSMVS executes the PROC named JS6PROC. The figure shows a listing of the JSMVS member.

### Sample JSMVS Member in SJSSAMP

```
//JSMVS JOB (ACCT),'PROGRAMMER NAME'
//*****
//* EXECUTE JCL CHANGE (MVS SYSTEMS)
//* REF: SAMPLIB (JSMVS)
//*
//* NOTE: MODIFY JOB STATEMENT AND ITEMS INDICATED BY ARROWS <==
//*****
// EXEC JS6PROC,PRM=R
//JLCHMO DD DISP=SHR,DSN=YOUR.OUTPUT.PDS <== WRITE CHANGED JCL
//JLRULES DD DISP=SHR,DSN=YOUR.CHANGE.CMDS (MBR) <== JCL CHANGE CMDS
//SYSIN DD *
INPUT SELECTION
DSN=YOUR.JCLLIB <== INPUT JCL LIBRARY
SELECT=YOURJOB <== MEMBER TO CHANGE
SOURCE TYPE=PDS <== TYPE OF INPUT LIB
//
```

### To run JCL Reformatter in batch mode

1. Copy and Edit the SJSSAMP library member JSMVS.
2. Modify the JOB statement according to site standards.
3. Change PRM=S to PRM=R in the EXEC statement.
4. If using Reformatter commands, modify the JLRULES DD statement to specify the library and specific member containing the commands to use for this JCL Reformatter run.
5. Edit the JLCHMO DD statement to specify the library where the reformatted JCL should be written. To perform a simulation, comment out this statement.



ASG *strongly* recommends *not* overwriting the input JCL, from the INPUT STATEMENT with the reformatted output JCL. Instead, give JLCHMO DD a different name and validate the reformatted JCL from this library.

6. Edit the Input Selection DSN= statement to specify the library containing the JCL members to reformat.
7. Edit the SELECT= statement to specify the members to change.
8. Edit the SOURCE TYPE= statement to identify the type of library containing the JCL. SOURCE TYPE=PROCLIB is not valid for REFORMAT.
9. Remove or comment out the Report Options statements.
10. Submit JSMVS for execution.
11. Review the held output for results.
12. Add Reformatter options by placing the BX Control Tables in the SYSIN control statements. Alternatively, Reformatter commands can be placed in a command file identified by a JLRULES DD statement.



- Commands in JLRULES take precedence over all Control Tables.
- The Control Table in effect is the last Control Table read in this order:
  - JSOPT02
  - JSOPT03
  - SYSIN in the JSMVS member.
- If JS6PROC is renamed, use the PROC Run Option to specify the new name of the PROC. See [Run Option Statements](#) for complete information about Run Options.
- JS6PROC provides the capability to change, reformat, or validate JCL in batch by coding:  

```
//S1 EXEC JS6PROC,PRM=xxx
```

 where xxx is any combination of these:
  - C to change JCL
  - R to reformat JCL
  - S to validate JCL

## Reformatting Comments with the BC Table

Reformatter provides multiple options setting up default reformatting for comment fields and comment statements.

This section describes reformatting both types of comments. Comment statements and fields in non-reformatted statements are not reformatted. The figure shows the placement of comment fields and comment statements. A comment field is all the text after the first blank following a JCL parameter or operator. A comment statement is any line that begins with the comment delimiter, typically `//*`.

A line of JCL contains an operator, parameter or a full statement of JCL.

### Sample JCL

```
//ASGG JOB (BNW),'JOB TYPE', COMMENT FIELD ON JOB
//      MSGLEVEL=(1,1), COMMENT FIELD FOR MSGLEVEL
//      MSGCLASS=X,
//      NOTIFY=ASGGDP,
//      REGION=200K
//* THIS IS A COMMENT STATEMENT
//STEP1 EXEC PGM=IEFBR14          COMMENT FOR THE STEP
//DD1   DD DSN=ASGUSR.JCL,DISP=(OLD,KEEP) COMMENT FOR DISP
//* THIS IS A COMMENT STATEMENT
```

Each comment option is specified in Control Table BC. (See [Control Table BC: Additional Reformatter Options.](#))

When acting on comment fields, JOB/SCAN Reformatter operates in one of three modes:

Mode	Description
Sentence	Refers to placing all comment fields for a JCL statement on consecutive lines, right justifying the text. This is the default mode.

Mode	Description
<b>Affinity</b>	Associates a comment field with the parameter it shares a line with or with the parameter on the line immediately preceding.
<b>Template</b>	Keeps comment fields in the same position, relative to the first line of the JCL statement and in the same column.


## Reformatting Comment Fields

### Repositioning Comment Fields

Comment fields can be repositioned by:

- moving to the top of the JCL statement using a sentence-like structure.
- having an affinity to a parameter on the same line.
- placing on the same relative line (template) as they were before reformatting. This table lists values for in column 20 of Control Table BC that reposition comment fields.

Value	Description
1	Moves all comment fields to the top of the JCL statement using a <b>sentence</b> -like structure. The comment field(s) are moved to the first line of the JCL statement and right-justified. As much of the comment as possible is placed on that line, with at least one blank after the last parameter on that line. The comment field(s) continue onto subsequent lines as necessary.
2	Tells JOB/SCAN that comment fields have an <b>affinity</b> to one of the parameters on the same line and move with that parameter. Parameters with comment fields will be reformatted to one parameter per line. (See <a href="#">Control Table BC: Additional Reformatter Options</a> for more information.)
3	Replaces comment fields on exactly the same relative line in exactly the same relative columnar position ( <b>template</b> ) as they were before reformatting. If the line of JCL is too long and would overlap the comment field, the line will not be reformatted and a message is issued in the output listing indicating that the line could not be reformatted. If the reformatted output contains fewer lines of JCL parameters than lines of comment fields, comment statements will be inserted to maintain the relative line position of the comment fields. A before-and-after example follows: <ul style="list-style-type: none"> <li>• Before: <pre>//DD1 DD DSN=A.B.C,          XXXX //          DISP=SHR, //          UNIT=A34          YYYY</pre> </li> <li>• After: <pre>//DD1 DD DSN=A.B.C,DISP=SHR,UNIT=A34  XXXX //*</pre> </li> </ul>

Value	Description
	//* <span style="float: right;">YYYY</span>
	<div style="border: 1px solid purple; padding: 5px; margin: 5px 0;">  To eliminate the intervening blank comment statements in the example, enter a value of 1 in column 31 of Control Table BC. </div>
N	Removes comment fields on all JCL statements. (This value does not remove <code>//*</code> comment statements.)

### Assigning Affinity Options Comment Fields

Use the values from this table in column 31 of Control Table BC to assign affinity to a comment field.

Value	Description
L	Specifies affinity to the last parameter on the line.
F	Specifies affinity to the first parameter on the line.
A	Specifies affinity to all parameters on the line. The comment field is replicated to follow each of the parameters.

### Assigning Template Options for Comment Fields

Use the values from this table in column 32 of Control Table BC to assign a start column value or left- or right- justification for comment fields.

Value	Description
M	(Default) Retains relative columnar position. If it overlaps a parameter, the comment field drops to the next line as a comment statement, retaining columnar position.
S	Specifies the column where the comment field is to begin. If it overlaps a parameter, the comment field slides to the right as much as necessary. If it does not fit, the comment field drops to the next line as a comment statement. Specify the beginning column number (two digits) in columns 33 and 34 of Control Table BC.
R	Right-justifies the comment field. The comment field drops to the next line as a comment statement, if necessary.
L	Left-justifies the comment field. The comment field drops to the next line as a comment statement, if necessary.



## Special Notes Regarding Comment Field Reformatting

Some combinations of entries in BC table column 20 and column 30 interact in notable ways:

- A 1 in column 20 and a T in column 30 instruct Reformatter to use sentence style for comment fields and to move an imbedded comment statement to the top of the JCL statement. In this case, Reformatter ignores the column 30 entry giving priority to arranging comment fields into sentence mode and places the comment statement after the JCL statement. Reformatter also treats an L in column 30 combined with a 1 in column 20 the same: arranging the comment fields in sentence mode takes priority and all interrupting comment statements are placed after the JCL statement.
- A 3 in column 20 and a T or L in column 30 instruct Reformatter to keep the comment fields in their same relative position on line and column (as though the comment fields were already in sentence mode) and either:
  - with T move an imbedded comment statement to the top of the JCL statement.

**Or**

- with L leave the imbedded comment where it is.

In either case, keeping comment fields in place is incompatible with leading-in or imbedded comment statements if JOB/SCAN re-orders parameters during reformatting.

- A 2 in column 20 and either T or L in column 30 instruct Reformatter to associate comment fields with a parameter on that line. In this case, moving the comment statement before the JCL statement for T, or leaving it on its original line for L, does not interrupt the flow of the comment fields.

## Reformatting Comment Statements (/\* Lines)

Imbedded comment statements can be directed either to move to the bottom of the JCL statement, to move to the top of the JCL statement, or to have an affinity to the last parameter on the preceding line. Use column 30 in Control Table BC. This table describes the values.

Value	Description
B	Moves all of the comment statements to the bottom of the JCL statement.
T	Moves all of the comment statements to the top of the JCL statement.
L	Tells JOB/SCAN that comment statements have an affinity to the last parameter on the preceding line. The comment statements move with that parameter.

## Keeping Comment Lines with a Parameter

This table shows the affinity options that are available for reformatting comment lines on JCL statements. Code the value in column 31 of Control Table BC.

Value	Description
L	Specifies affinity to the last parameter on the line.
F	Specifies affinity to the first parameter on the line.
A	Specifies affinity to all parameters on the line. The comment line is replicated to follow each of the parameters.

This table shows the columnar options that are available for reformatting comment fields on JCL statements. Code the value in column 32 of Control Table BC.

Value	Description
M	(Default) Retains relative columnar position. If it overlaps a parameter, the comment field drops to the next line as a comment statement, retaining columnar position.
S	Specifies the column in which to begin the comment field. If it overlaps a parameter, the comment field slides to the right as much as necessary. If it does not fit, the comment field drops to the next line as a comment statement. Specify the beginning column number (two digits) in columns 33 and 34 of Control Table BC.
R	Right-justifies the comment field. The comment field drops to the next line as a comment statement, if necessary.
L	Left-justifies the comment field. The comment field drops to the next line as a comment statement, if necessary.

## Removing Sequence Numbers from Columns 73-80

### To remove the sequence numbers

- Use this command in a reformat member:

```
ON
  JCL UNNUM
END
```

### Or

- Specify this Control Table BC entry in JSOPT02 to remove the sequence numbers:

```
.....+.....1.....+.....2.....+.....3.....+.....
      BC00000,00000
```

## Commenting a Reformat Member

Add comments to a reformat member (a dataset member that contains change commands) using the delimiter `/*`. There is no ending delimiter. The figure shows examples of comments in a reformat member.

### Comments in a JCL Change Member

```
/* THIS IS A FULL LINE COMMENT.  THE DELIMITER STARTS IN COL. 1  
ON JOB  
  PLACE CLASS ON LINE 2    /*THIS IS A PARTIAL LINE COMMENT  
  PLACE TIME ON LINE 3    /*HERE IS ANOTHER PARTIAL LINE COMMENT  
END                        /*END THE COMMAND
```

# ROSCOE Interface

This topic provides information about using JOB/SCAN in a ROSCOE operating environment.

- [ROSCOE Interface Facilities](#)
- [ROSCOE RPF Facility and ISPF Functional Differences](#)
- [Execution of JOB/SCAN RPF Panels](#)
- [ROSCOE RPF Panels](#)
- [ETSO Overview](#)
- [ETSO RPFs](#)

## ROSCOE Interface Facilities

In the ROSCOE RPF environment, the ASG RPF facility provides RPF panels similar in organization to the ISPF panels. Use these panels to select reports to create and display, and jobs to be scanned. Creation of reports and scanning of jobs are not available in foreground mode. However, this is accomplished by selecting batch execution via the panels. ROSCOE users can specify a ROSCOE library type in addition to other valid library types. This enables JOB/SCAN to accept JCL as input from a ROSCOE library.

The ROSCOE ETSO environment extends the capabilities of JOB/SCAN under ROSCOE. When utilizing ROSCOE's ETSO interface, JOB/SCAN is executed under ROSCOE with all the functions of the TSO/ISPF environment. Operation under ROSCOE ETSO includes foreground execution.

### **SJJSRRPF and SJJSRTSO Libraries**

These libraries contain RPFs, panels, and options used during installation of JOB/SCAN in a ROSCOE environment:

- The SJJSRRPF library provides the RPFs and panels necessary to run JOB/SCAN in a ROSCOE RPF environment without ISPF.
- The SJJSRTSO library provides the RPFs necessary to run JOB/SCAN in the ROSCOE ETSO environment and ISPF ROSCOE ETSO environment.



You must install the appropriate libraries if you intend to run JOB/SCAN under ROSCOE. Refer to [ROSCOE Interface](#).

### **RPF Overview**

The ROSCOE RPF Facility provides a set of ROSCOE RPFs and panel definitions that enable execution of JOB/SCAN within the ROSCOE environment. The RPFs and panels are contained in a separate library on the JOB/SCAN distribution tape. It is unloaded at installation time into a PDS library called SJJSRRPF.

The library contains an RPF to copy all of the RPFs and panels to a system ROSCOE library (JSRUNLD). Once copied, one permanent variable (P16) is modified to point to the ASG libraries.

The JOB/SCAN ROSCOE panels and functions are similar to the ISPF version. However, there are differences because of ROSCOE constraints. These differences are described in the next section.

In general, the ROSCOE panel formats and functions follow the documentation for the ISPF panels.

### JOB/SCAN Library Member Considerations

There are no special modifications necessary to any of the entities such as JS6PROC or SJJSPARM library members. These should be set up and installed, as they would be in a non-ROSCOE environment.

The ROSCOE panels submit JOB/SCAN batch jobs. The panels do not provide execution in foreground. Therefore, the SJJSPARM library information pertaining to foreground execution is ignored. The ROSCOE RPFs and batch jobs that are submitted use the SJJSPARM library members for run control and library allocations. The members located in the SJJSPARM library are used, unless a user has an SJJSPARM member of the same name in their own ROSCOE library, in which case that member is used.

## ROSCOE RPF Facility and ISPF Functional Differences

Follow all steps in [Installation](#) for loading libraries. ROSCOE RPFs and panels are loaded during the installation process.



There are approximately 50 RPFs and 3,000 lines of RPF code in the library.

The following functions are available with the ROSCOE panels but are not included with the ISPF panels:

- ROSCOE users can elect to 'ATTach' the JOB/SCAN JCL that was built by the ROSCOE panels by selecting the 'A' option on the panel. This enables the JOB/SCAN JCL to be viewed and/or edited prior to submission.
- ROSCOE users can specify a library type of 'AWS', 'ROS' or 'ROSCOE' in addition to other valid library types. This enables JOB/SCAN to accept JCL as input from a ROSCOE library.



Only one member may be specified in the SELECT MEMBER(S) field of the panel as MEMNAME or PREFIX.MEMNAME. This JCL member is copied into the JOB/SCAN executable JCL following this statement:

```
//JLJOB DD DATA,DLM=###,DCB=BLKSIZE=80
```

If additional members are desired, the 'A' option should be selected from the panel and the desired members may be copied, via ROSCOE commands, into the JOB/SCAN JCL following the DD DATA card.

- All ROSCOE panels may be exited by entering an 'X' in the option field or by pressing <PF3> or <PF15>. You may also type 'CLEAR' to exit RPFs.

The following functions are not available with the ROSCOE panels:

- Foreground execution of JOB/SCAN ROSCOE panels provide for submission of batch jobs. For more information about batch processing, see [Validating JCL in Batch](#) or [Running Reformatter in Batch Mode](#).

- Product Administrator options are not available. This means that the ASG-JOB/SCAN Primary Menu cannot be tailored based on USERID, and the SJJSPARM library members cannot be edited from the panels.

Variable panel options are set up that can be used as indicated earlier. If an alternate primary menu configuration is desired, a new panel must be created and the RPF JSRMAIN must be modified.

Other functions not available with the ROSCOE panels are:

- Assign Alternate Options Members
- Help panels
- Limited directory support (display only, no processing)

## Execution of JOB/SCAN RPF Panels

Make sure to set up members JSOPT01 and JSOPT02 as outlined in [Installation](#) before the JOB/SCAN RPF panels are executed.

To execute the JOB/SCAN RPF panels, enter the command JSRMAIN.

If this is the first-time these panels have been executed, the first display is an introductory panel. Press <Enter> to continue, or press <PF3> or <PF15> to terminate.

Make the requested entries into the JOBCARD panel. The jobcard is saved by the RPFs when building batch runs of JOB/SCAN. When finished, enter an 'X' in the option field, or press <PF3> or <PF15> to exit.

The next panel is the Library Specification panel. Enter the fully-qualified dataset name where requested for 'Input library for member selection'. Enter the library type where requested. Valid entries are PDS, PAN, PANVALET, LIB-3, LIBRARIAN-3, LIB, LIBRARIAN, SEQ, SEQUENTIAL, PROC, and PROCLIB. Additionally, AWS, ROS, and ROSCOE are valid entries from the ROSCOE RPF panels. Enter an 'X' in the option field or press <PF3> or <PF15> to exit.



See the ISPF documentation for entries in this panel. In the ROSCOE RPF panels, specify fully-qualified dataset names without quotes.

## ROSCOE RPF Panels

The panel examples on the following pages are typical of the RPF panels the ROSCOE user sees when running JOB/SCAN under ROSCOE RPF.

### ASG-JOB/SCAN Primary Menu

```

----- ASG-JOB/SCAN PRIMARY MENU -----
SELECTION ==>>

  0 - Hardcopy Output and JOB Statement Definition      USERID - CS081A
  1 - Validate JCL                                     PREFIX  - CS081A
  2 - Change / Reformat JCL                           DATE    - 09/20/2016
  3 - Processing Options                               TIME    - 09:58
  4 - Selection List Maintenance                       APPLID  - JS
  X - Exit

```

## Hardcopy Output Characteristics Panel

```

----- HARDCOPY OUTPUT CHARACTERISTICS ----- ( 1 of 2 )
COMMAND ==>

Force OPE hardcopy for Initialization ==>   (Y/N)

Enter/verify values for Hardcopy output:
Class      ==>
Dest       ==>
Copies     ==>          1-255 (JES2) or 1-254 (JES3)
UCS        ==>          (For other than IBM 3800)
Chars      ==>          (Only for IBM 3800)
Forms      ==>
Printer    ==>          (3800/9700/2700/IMPACT)
Hold       ==>          (Y/N)
Subsystem  ==>          (JES2/JES3/NONE)

If you wish to be prompted with this panel when the Hardcopy option
is specified for foreground options, enter Y below:
Hardcopy panel prompt ==>   (Y/N)

Press <END> for next panel

```

## Job Statement for Batch Submission Panel

```

----- JOB STATEMENT FOR BATCH SUBMISSION ----- ( 2 of 2 )
COMMAND ==>

JOB STATEMENT INFORMATION: (Required - Enter/Verify JOB control statements)
==>
==>
==>
==>
==>

Step accounting information for generated EXEC statement (61 chars max)
ACCT ==>

If you wish to be prompted with this panel when the Batch option
is specified, enter Y below:
JOB Statement panel prompt ==>   (Y/N)

Press <END> to return.

```

## Validate JCL Panel

```

----- JOB/SCAN - JCL VALIDATION -----
COMMAND ==>

```

```

      F - Foreground execution mode      P - Processing Options
      B - Batch execution mode          E - Edit JCL member
                                       BR - Browse JCL member

Input Library  ===> SYSUSR.JCL
Input Type    ===> J                    (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ===> PDS                  (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ===> LEDGR0              (Single member if BATCH with TRANSMIT)
Exclude Members ===>                    (Not valid for BATCH with TRANSMIT)
First Proclib ===>
Proclib Type  ===>                      (PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
Report Heading ===>
Display       ===> Y (Yes/No/ErrorsOnly) Browse/View ===> B (B/V)
Hardcopy      ===> N (Yes/No/ErrorsOnly/SpaceOnly/JobErrors)
Options in Effect Listing ===> N (Y/N)   SJL PDSOUT ===> N (Y/N)

Selection Exit ===>                      Parm ===>

```

## Change/Reformat Panels

```

----- JOB/SCAN - CHANGE / REFORMAT JCL -----
COMMAND ===>

      A - ATTACH JOB/SCAN JCL before SUB  B - SUBmit job (Do not Attach JCL)
      X - Exit

Command Lib    ===>
Member Name   ===>

Input Library  ===>
Input Type    ===>                    (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ===>                    ( AWS, ROS, PDS, PAN, LIB, LIB-3, or SEQ)
Select member(s) ===>
Exclude member(s) ===>

Change JCL    ===>                    (Y/N)
Reformat JCL  ===>                    (Y/N)
Validate JCL  ===>                    (Y/N)

Report Heading ===>
              ENTER Continue, PF#/PF15 Primary Menu, CLEAR Exit RPFs

```



```

----- JOB/SCAN - CHANGE / REFORMAT JCL -----
COMMAND ==>

Write Changed JCL? ==>   (Y/N)

Output Lib   ==>
Library type ==>         (PDS, PAN, LIB, LIB-3, or SEQ)

JCL Change Report:
  Display    ==>         (Out JCL/Audit/None)
  Hardcopy   ==>         (Out JCL/Audit/None)

Processing Summary:
  Display    ==>         (Y/N)
  Hardcopy   ==>         (Y/N)

ENTER Continue, PF#/PF15 Primary Menu, CLEAR Exit RPFs

```

```

----- JOB/SCAN - VALIDATE CHANGED/REFORMATTED JCL -----
COMMAND ==>

First Proclib   ==>
  Proclib Type ==>         (ROS, PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
  Display        ==>         (Yes/No/ErrorsOnly)
  Hardcopy       ==>         (Yes/No/ErrorsOnly/SpaceOnly)

Options in Effect Listing ==>   (Y/N)

Selection Exit ==>           Parm ==>

Standards Name ==>           Parm ==>

ENTER Continue, PF#/PF15 Primary Menu, CLEAR Exit RPFs

```

### Selection List Panels

```

----- JOB/SCAN SELECTION LIST MAINTENANCE FACILITY -----
COMMAND ==>

      C - Create a Selection LIST
      M - Modify a Selection LIST

LIST Library    ==>
LIST Type       ==>         (AWS, ROS, OR PDS)

```

```

LIST Name      ===>                (Leave blank for directory list)

                                * (LIST Library required for PDS)

Use ROSCOE commands to Delete, Copy, Print, and Rename LISTS

                                ENTER Continue, PF#/PF15 Primary Menu, CLEAR Exit RPFs

```

```

----- CREATE A SELECTION LIST -----
COMMAND===>

                                LIST          AWS
LIST Description  ===>

JCL Library name  ===>
JCL Library type  ===>                (PDS, PAN, LIB, LIB-3)

Enter SELECT= detail level statements:

SELECT=
SELECT=
SELECT=
SELECT=
SELECT=
SELECT=
SELECT=
SELECT=
SELECT=
SELECT=
SELECT=

                                Use Modify panel to add additional lines to this list

                                ENTER Continue, PF#/PF15 Primary Menu, CLEAR Exit RPFs

```

## ETSO Overview

The ROSCOE ETSO facility provides a set of user written RPFs and ROSCOE support members. This enables the interactive execution of JOB/SCAN and REFORMATTER, and the ability to invoke the JOB/SCAN ISPF panels via ROSCOE. The RPFs and support members are contained in a separate library supplied on the JOB/SCAN ship tape, which is unloaded at installation time into a PDS library called SJSRSTSO.

### The SJSRSTSO Library

The SJSRSTSO library contains ROSCOE RPFs and ETSO installation support members for the interactive execution of JOB/SCAN ISPF dialog.

Library Members	Description
JSETSOPG	Contains the entries required for inclusion in the ROSCOE Eligible Program List for releases prior to 5.7 of ROSCOE
JSETSOP7	Contains the entries required for inclusion in the ROSCOE Eligible Program List for Release 5.7 (or later) of ROSCOE
JSCAN	Enables invoking JOB/SCAN interactively via ROSCOE
JREF	Enables invoking REFORMATTER interactively via ROSCOE
JSRISPF	Used to invoke the JOB/SCAN ISPF dialog



When running JOB/SCAN APF-authorized through ROSCOE, be sure the name J02YJS is added to the ROSCOE Eligible Program List.

#### JOB/SCAN SJJSPARM Library Member Considerations

There are no special modifications necessary to any of the normal JOB/SCAN entities such as JS6PROC and the SJJSPARM library members. The exception is JSOPT02 that uses &UID as part of the dataset name. To conform to site standards, it may be necessary to modify any statement in JSOPT02 that uses &UID as part of a dataset name.



The changes indicated here are only necessary if the ETSO feature is to be used.

Library Members	Description
JSOPTRS	This member provides required control cards for the interactive execution of JOB/SCAN. It resides in the SJJSPARM library. This member provides the same control capabilities for the JSCAN RPF that JSOPTM provides for the JSCAN edit macro under ISPF. Refer to <a href="#">Reference Information</a> for information about JSOPTM.

#### ROSCOE ETSO Facility and ISPF Functional Differences

The following ISPF type functions are available with ROSCOE ETSO.

- There is a JSCAN and JREF Edit Macro look-alike. A JSUB can be built using the JSCAN RPF and the ROSCOE variable S.RC.
- The ISPF panels are the same ISPF panels used under TSO and have all the same functions and capabilities as when invoked via TSO/ISPF/PDF.

The following ISPF functions are *not* available with ROSCOE ETSO:

- There is no JCHG, JVER, JEM, or JSUB Edit Macro facility, except as noted above.
- Currently the JOB/SCAN ISPF panels cannot access a ROSCOE library directly. JCL must reside in a PDS, PANVALET, LIBRARIAN, SEQUENTIAL, or PROCLIB type dataset.

## ETSO RPFs

This topic provides information on the usage of ETSO RPFs.

### JREF

With JCL in AWS, enter JREF on the command line. This invokes REFORMATTER interactively through ROSCOE.

The JREF RPF reads the JCL in AWS as input, processes the JCL, and rewrites the reformatted JCL back to AWS, replacing the original JCL. Save or delete the JCL.

### JSCAN

With JCL in AWS, enter JSCAN on the command line. This invokes JOB/SCAN interactively via ROSCOE.

The JSCAN RPF reads the JCL in AWS as input to a JOB/SCAN execution. The RPF builds a dataset in which JOB/SCAN writes the Structured JCL Listing. The RPF attaches this dataset for display of the Structured JCL Listing. ROSCOE commands and PF keys may be used for navigation through the report. The ROSCOE POINT command may be used to locate a specific message identified in the Error Summary.

To terminate viewing of the Structured JCL Listing, enter GO on the command line to return to the JCL in AWS.



The dataset built during execution of JSCAN remains until the next execution of JSCAN.

Currently this RPF cannot scan PROCS in AWS unless this PROC is set up as an instream PROC with a JOBCARD, PEND, and EXEC statements.

### JSRISPF

The JSRISPF RPF is used to invoke the JOB/SCAN ISPF dialog. This is performed by entering JSRISPF on the command line, or another name if one was given to this RPF when it was installed. The RPF allocates all of the required ISPF support libraries to initialize an ISPF application under ROSCOE. The RPF then issues a call to the program that initializes the JOB/SCAN ISPF dialogue.

ISPF is now executing from ROSCOE. All functions and PF key assignments are ISPF/PDF.

# User Information - JOB/SCAN with Schedulers

This topic provide information about the use, setup and installation of the interface between JOB/SCAN and following schedulers:

- [CA-7](#)
- [ESP](#)
- [TWS](#)

# User Information - JOB/SCAN with CA-7

This documentation describes the interface between ASG-JOB/SCAN (herein called JOB/SCAN) and the CA-7 Scheduler. It explains how to initialize and execute the interface, and provides Edit macro and batch mode instructions.

Topic	Description
<a href="#">Getting Started</a>	Introduces the facilities of the CA-7 interface.
<a href="#">Using JCA to Validate JCL</a>	Describes the Edit Macro validation function.
<a href="#">Validating Jobs from Foreground</a>	Describes how to resolve variables and validate JCL through the ISPF panels.
<a href="#">Using CA7SCHD to Validate a CA-7 Schedule</a>	Describes validating schedules through batch.
<a href="#">Using JSCA7 to Validate a JOB</a>	Describes validating a member so that variables are resolved.
<a href="#">JOB/SCAN CA-7 Interface Setup</a>	Describes the installation of the interface.

## Getting Started

ASG provides this handbook to IT professionals who work with JCL members containing CA-7 #Jx control statements and CA-Driver procedures. The crucial need is for CA-7 and CA-Driver to resolve the CA-7 control statements in the JCL before validation by JOB/SCAN.

The facilities provided through JOB/SCAN support these requirements, enhancing the overall quality and efficiency of the JCL processing at sites using CA-7 Job Management and CA-Driver software.

### Usage Scenarios

This JOB/SCAN CA-7 scheduler interface resolves CA-7 #Jx control statements and CA-Driver variables before JOB/SCAN validation. The JOB/SCAN CA-7 interface has two approaches to validating JCL:

- Use the material in this documentation when you have members that contain both CA-Driver and CA-7 variables and want to validate them through batch, ISPF panels, or Edit macro modes.
- Use the material in [Scheduler Interfaces](#) when you have legacy batch processes utilizing the previous JOB/SCAN-CA-7 interface. These remain supported without modification to your supporting jobs. See [CA-7 Scheduler Interface](#) for information about this interface.

### Terminology

The following terms are used in this documentation:

#### CA-7 # Control Statements

The statements that CA-7 interprets to dynamically change job stream contents. Among these are #ARF, #HLD, #JI, #JO, #MNT, #XI, and #XO.

## CA-Driver Procedures

The procedures CA-Driver expands into the job stream at job submission time.

### Catalog (Run-Time) Simulation

Thorough validation of production JCL requires that the JCL be analyzed on multiple levels to identify potential problems:

- On the most basic level, JOB/SCAN ensures that the JCL is *syntactically* correct.
- At the *run-time* level, JOB/SCAN identifies:
  - environmental errors, such as security violations and programs not found.
  - dataset disposition errors.

Whether processing a single JCL member or a sequence of members, JOB/SCAN uses its powerful z/OS *catalog simulation* facility to ensure that the JCL steps will create, use and delete datasets as expected.

See [User Information - JOB/SCAN](#) for a thorough description of JOB/SCAN's z/OS simulation.

## Using JCA to Validate JCL

JOB/SCAN's Edit macro takes the JCL in the Edit workspace, gives it to CA-7 to process and then validates the resulting JCL.

### JCA Edit Macro

The purpose of the Edit macro is to validate a single member. JCA supports these special command-line parameters:

Parameter	Description
SJOB	Member name under which JOB/SCAN stores the JCL in the CA-7 JCL library.
SDATE	Scheduled Run date; default is current date. Format is: YYMMDD
STIME	Scheduled Run time; default is current time. Format is: HHMM
SCHID	Three character schedule ID; default is 001.
DEBUG	Displays temporary files as they are used.
RESET	Resets the Schedule Criteria panel for display (unsuppresses it).

For more information about other parameters for JOB/SCAN's Edit macros, see [User Information - JOB/SCAN](#) or [Reference Information](#).



Before using any function in this documentation, ask your administrator to be sure to follow installation and activation instructions in [JOB/SCAN CA-7 Interface Setup](#).

## Using the JCA Edit Macro in ISPF Edit or View

To use this method, you Edit/View a member in an ISPF EDIT/VIEW session. To achieve the most accurate validation of JCL members with JOB/SCAN, the CA-7 # control statements must be resolved before validation. The JCA Edit macro first resolves CA-7 # control statements and CA-Driver procedures, then validates the JCL.

### To validate JCL from within an EDIT/VIEW session

1. Bring the JCL member into ISPF EDIT.
2. Type JCA and any optional parameters on the command line. JOB/SCAN displays the CA-7 Schedule Criteria panel.

### Invoking the JOB/SCAN CA-7 interface with the JCA Edit macro

```

EDIT          SYSUSR.JCL(TEST) - 01.00                Columns 00001 00072
Command ==> JCA stime(1220)                          Scroll ==> PAGE
000001 //PC28002A JOB (PRV1PD,132900,DSS,37),PC28002,MSGCLASS=X
000002 /*ROUTE PRINT PRV103
000003 //S1      EXEC PGM=IEFBR14, !:20021(&3:0000)
000004 //              REGION=
000005 #JI,SCHID=002
000006 //DD1      DD DSN=PRV1.SMS.PC28002A,
000007 //              DATACLAS=SRCFLIB,
000008 //              LRECL=80,RECFM=FB
000009 /** !:29936 !:29937 !:20097 !:20112 !:20053 !:20054
000010 //DD2      DD DSN=PRV1.SMS.PC28002B,
000011 //              DATACLAS=SRCFLIB,
000012 //              LRECL=80,RECFM=FB
000013 /** !:29936 !:29937 !:20097 !:20112 !:20053 !:20054
000014 #JEND
000014 #JO,SCHID=002
.....

```

3. Input member or JOB name, schedule ID and simulation date and time into the fields.

### Specifying parameters for the JOB/SCAN CA-7 interface

```

- JOB/SCAN ----- CA-7 SCHEDULE CRITERIA -----
COMMAND ==>

SPECIFY SCHEDULE SIMULATION INFORMATION

MEMBER/JOB NAME      ==> TEST
SCHEDULE ID         ==> 001

SIMULATION DATE      ==> 12/07/05   (MM/DD/YY)
SIMULATION TIME      ==> 11:13     (HH:MM)

_ To suppress this panel in the future.

REVIEW VALUES AND PRESS ENTER TO CONTINUE OR END TO RETURN.

```

Field	Description
MEMBER/JOB NAME	The name of the Job. The default value is the member name in Edit or View.



Field	Description
SCHEDULE ID	A valid three-digit schedule ID; 001 is the default.
SIMULATION DATE	A valid date. Current date is the default.
SIMULATION TIME	A valid time. Current time is the default.
Suppress	Mark with an X to suppress the displaying of this panel. Use the RESET parameter with the JCA command to redisplay this panel.

4. Press Enter. JOB/SCAN displays the SJL report.

## Validating Jobs from Foreground

You can use JOB/SCAN's foreground processing to validate jobs that use the CA-7 variables.

### Set Up to Use Foreground Validation



Before using any function in this documentation, ask your administrator to be sure to follow installation and activation instructions in [JOB/SCAN CA-7 Interface Setup](#).

### To set up JOB/SCAN to validate JCL from foreground

1. Set the Scheduler detail statement of the Run Options statement in JSOPT02 as shown in the below figure.

#### Setting the Run Options for the CA-7 scheduler interface

```

** JSOPT02  ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** *
**
** JSOPT02 OPTIONS MAY BE OVERRIDDEN BY CORRESPONDING OPTIONS **
** CODED IN JSOPT03 OR JSOPTM. **
** **
...
RUN OPTIONS
  SCHEDULER=(CA7,SCAN)
...

```

2. Turn on scheduler processing:
  - a. Select JCL Validation panel command P.

#### Turning on Scheduler Processing

```

D02PVAL ----- ASG-JOB/SCAN - JCL VALIDATION -----
COMMAND ==> P

      F - Foreground execution mode          P - Processing Options
      B - Batch execution mode              E - Edit JCL member
      S - Submit After Validation           BR - Browse JCL member
Input Library ==> CS082A.JCL
...

```

- b. Type Y in the Use SCHEDULER interfaces field.

#### Turning on Scheduler Processing

```

D02PEP ----- ASG-JOB/SCAN PROCESSING OPTIONS -----
COMMAND ==>

  Use Error Table           ==> N           (Y/N)
  Condition Code Processing ==> N           (Y/N)
  Catalog Access           ==> Y           (Y/N)
  VTOC Access              ==> Y           (Y/N)
  PDS Member Verification  ==> Y           (Y/N)
  Security Exit DSNAME Check ==> N         (Y/N)
  Program Checking         ==> Y           (Y/N)
  Disable SMS Processing   ==> N           (Y/N)
  IDCAMS Processing        ==> Y           (Y/N)
  Use SCHEDULER interface  ==> Y           (Y/N)
  ...

```

- c. Press END to return and enable scheduler interfaces.

### Validating Members

#### To validate member in foreground

1. Set up JOB/SCAN according to [Set Up to Use Foreground Validation](#).
2. At the JCL Validation panel, use the F command.

#### Turning on Scheduler Processing

```

D02PVAL ----- ASG-JOB/SCAN - JCL VALIDATION -----
COMMAND ==>  F

  F - Foreground execution mode      P - Processing Options
  B - Batch execution mode           E - Edit JCL member
  S - Submit After Validation        BR - Browse JCL member
Input Library  ==> CS082A.JCL
Input Type    ==> J                  (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ==> PDS                (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> BR1*
Exclude Members ==>
First Proclib ==>
Proclib Type  ==>                    (PDS, PAN, LIB, or LIB-3)
  ...

```



If you input more than one member name (with list, range, and/or masks) JOB/SCAN displays the CA-7 Schedule Criteria panel for each one.

3. Press Enter. JOB/SCAN displays the CA-7 Schedule Criteria panel.
4. Input member or JOB name, schedule ID and simulation date and time into the fields.

### Specifying parameters for the JOB/SCAN CA-7 interface

```

D00PJCA1 /SCAN ----- CA-7 SCHEDULE CRITERIA -----
COMMAND ==>

SPECIFY SCHEDULE SIMULATION INFORMATION

MEMBER/JOB NAME      ==> BR14JOB
SCHEDULE ID         ==> 001

SIMULATION DATE      ==> 08/04/29   (YY/MM/DD)
SIMULATION TIME      ==> 17:49     (HH:MM)

_ Enter "X" to suppress this panel in the future.

Review values and press ENTER to continue or END to return.

```

where:

Field	Description
MEMBER/JOB NAME	The name of the job. The default value is the member name from the Select Members field from the JCL Validation panel.
SCHEDULE ID	A valid three-digit schedule ID; 001 is the default.
SIMULATION DATE	A valid date. Current date is the default.
SIMULATION TIME	A valid time. Current time is the default.
Suppress	Mark with an x to suppress the displaying of this panel.



Use the Reset parameter of the JCA command to redisplay this panel.

5. Press Enter. JOB/SCAN displays the SJL report.

## Using CA7SCHD to Validate a CA-7 Schedule



Before using any function in this documentation, ask your administrator to be sure to follow installation and activation instructions in [JOB/SCAN CA-7 Interface Setup](#).

The batch job CA7SCHD:

- Retrieves a list of jobs from CA-7 based on your forecast parameters.
- Extracts the JCL and prepares it for validation.

- Executes JOB/SCAN to validate the jobs in run-time order.

**To set up the batch job modify these statements in the CA7SCHD member in the SJJSSAMP library**

1. JOB card. Modify according to local preferences and requirements.
2. CA-7 generic user ID. This might be optional; consult with local system administration.
3. CA-7 password for the ID. If the generic user ID is optional, so is this password.
4. SASSBSTR parameter.
5. Dataset name of the JOB/SCAN SJJSCST library.
6. Dataset name for the CA-7 Command file. Replace `XXXX.XXXX.COMMAND` with the actual dataset name.
7. Alternate CA-7 Command file. If this DD is used, then enter `NULLFILE` in step 6.
8. CA-7 batch input files; use `NULLFILE` if pooling is used. See .
9. CA-7 batch output files; use `NULLFILE` if pooling is used. See .
10. The unit name for new datasets.
11. The release number for CA7. If the number is less than 11 JOB/SCAN expects the CA-7 release 10 format for the FJOB report; otherwise, it expects release 11 format.
12. Provide the location of the JS6PROC (if it is not in an accessible PROCLIB) by uncommenting this statement and replace `XXXX` with the actual node.
13. Modify or add forecast parameters to define the jobs you want to have validated.
14. (Optional) To provide the location of the CA-7 LOAD library, uncomment this statement and replace `CA7 LOAD LIBRARY` with the actual dataset name of the library.
15. Replace the `@` with a digit from 1-9 in the DD name only if line 8 contains a dataset name for the batch input file.
16. Replace the `@` with a digit from 1-9 in the DD name only if line 9 contains a dataset name for the batch output file.

### Modifying the CA7SCHD member for processing a CA-7 schedule

```

1. //CA7SCHD JOB (ACCT), 'PROGRAMMER NAME', NOTIFY=&SYSUID, REGION=6M <==
//*****
//* INTERFACE TO CA-7 SCHEDULER & CA-DRIVER
//* REF: SJJSSAMP(CA7SCHD)
//*
//* NOTE: MODIFY JOB STATEMENT AND ITEMS INDICATED BY ARROWS <==
//*****
2. // SET ID= <== GENERIC USER
3. // SET PASSWORD= <== PASSWORD
4. // SET PARM= <== SASSBSTR PARM
5. // SET EXECLIB=XXXX.XXXX.SJJSCLST <== ASG CLIST LIBRARY
6. // SET CA7CMDS=XXXX.XXXX.COMMAND <== CA-7 COMMAND FILE
7. // SET UCC7CMDS=NULLFILE <== ALTERNATE CA-7 CMD
8. // SET BATCHIN=NULLFILE <== CA-7 BATCH INPUT FILE
9. // SET BATCHOUT=NULLFILE <== CA-7 BATCH OUTPUT FILE
10. // SET SYSDA=SYSDA <== VALID UNIT
11. // SET CA7REL=11 <== CA-7 RELEASE LEVEL
//*
12. //*PROCLIB JCLLIB ORDER=DSSI.XXXX.SJJSSAMP <== FOR JS6PROC (UNCOMMENT)
//***** WRITE CA-7 FJOB CONTROL STATEMENT TO A TEMPORARY DATA SET
//*
//SAVECTL EXEC PGM=IRXJCL, PARM='J00CCTL &ID &PASSWORD'
//*-----*
//* MODIFY THE FOLLOWING FJOB STATEMENT TO DEFINE THE SCHEDULE *
//* TO BE VALIDATED BY JOB/SCAN: <==MODIFY THE FOLLOWING FJOB *
//* STATEMENT OR ADD STATEMENTS AS REQUIRED: *
//CCTLIN DD *
13. FJOB, SYS=GG, FROM=(MMDDYY, HHMM), TO=(MMDDYY, HHMM)
//*-----* <== MODIFY PREVIOUS-*
...
//SASSBSTR EXEC PGM=SASSBSTR, &PARM
14. //*STEPLIB DD DSN=CA7 LOAD LIBRARY, DISP=SHR <== CA7 LOAD (UNCOMMENT)
//CA7CMDS DD DSN=&CA7CMDS, DISP=SHR CA-7 CMD FILE
//UCC7CMDS DD DSN=&UCC7CMDS, DISP=SHR CA-7 CMD FILE
15. //BTERM0@I DD DSN=&BATCHIN, DISP=SHR <== CA-7 BATCH INPUT FILE
16. //BTERM0@O DD DSN=&BATCHOUT, DISP=SHR <== CA-7 BATCH OUTPUT FILE
...

```

17. Save and run the batch job by the usual means of submitting batch validation jobs at your site.
18. Use your site's usual method of printing or saving output for batch jobs (for example, the SDSF output queue) to review the results of the JOB/SCAN validation.

## Using JSCA7 to Validate a JOB

The batch job JSCA7 performs these tasks:

- Retrieves a job from the specified JCL Library.
- Interfaces with CA-7 to resolve statements.
- Executes JOB/SCAN to validate.

### Set up the Batch Job for a Single Member



Before using any function in this documentation, ask your administrator to be sure to follow installation and activation instructions in [JOB/SCAN CA-7 Interface Setup](#).

Modify these statements in the JSCA7 member in the SJJSSAMP library:

1. JOB card. Modify according to local preferences.
2. CA-7 JCL library dataset name. Replace the `XXXX.XXXX.XXXX` with the actual dataset defined in [JOB/SCAN CA-7 Interface Setup](#).
3. Dataset name for the CA-7 Command file. Replace `XXXX.XXXX.COMMAND` with the actual dataset name.
4. Alternate CA-7 command file. Replace `NULLFILE` with the actual dataset name, and place `NULLFILE` in the dataset name in step 3.
5. CA-7 batch input file. Use `NULLFILE` if pooling is used. See [Setting up Pooling \(Optional\)](#).
6. CA-7 batch output file. Use `NULLFILE` if pooling is used. See [Setting up Pooling \(Optional\)](#).
7. The unit name for new datasets.
8. (Optional) The location of the JS6PROC (if not in an accessible PROCLIB). Replace `xxxxxxx` with the actual dataset name and uncomment the statement.
9. Replace the `@` with a digit from 1-9 in the DD name only if line 5 contains a dataset name for the batch input file.
10. Replace the `@` with a digit from 1-9 in the DD name only if line 6 contains a dataset name for the batch output file.
11. CA-7 parameters. Use the comments in the next several lines as a guide to providing parameters to CA-7.
  - A valid schedule ID.
  - A valid JOB name.
  - A valid date.
  - A valid time.

For example:

```
CA7 SCHID(101) -
    SJOB(MYJOB) -
    SDATE(080501) -
    STIME(2153)
```

12. JOB/SCAN parameters. Place the dataset name, member, dataset type.

### Run the Batch JOBS

Use the usual means of submitting batch validation jobs at your site.

Use your site's usual method of printing or saving output for batch jobs (for example, the SDSF output queue) to review the results of the JOB/SCAN validation.

## Modifying the JSCA7 member for processing one JCL member at a time

```

1. //JSCA7 JOB (ACCT),'PROGRAMMER NAME'
   //*****
   /** EXECUTE JOB/SCAN (MVS SYSTEMS) WITH CA-7 BATCH PROCESSING
   /** REF: SJJSSAMP(JSCA7)
   /**
   /** NOTE: MODIFY JOB STATEMENT AND ITEMS INDICATED BY ARROWS <==
   /** REXX EXEC J00CAVAL SHOULD BE MODIFIED BEFORE RUNNING THIS JOB **
   //*****
2. // SET CA7LIB=XXXX.XXXX.XXXX <== CA-7 JCL LIBRARY NAME
3. // SET CA7CMDS=XXXX.XXXX.COMMAND <== CA-7 COMMAND FILE
4. // SET UCC7CMDS=NULLFILE <== ALTERNATE CA-7 CMD
5. // SET BATCHIN=NULLFILE <== CA-7 BATCH INPUT FILE
6. // SET BATCHOUT=NULLFILE <== CA-7 BATCH OUTPUT FILE
7. // SET SYSDA=SYSDA <== VALID UNIT
8. /**PROCLIB JCLLIB ORDER=XXXXXX.PROCLIB <== FOR JS6PROC (UNCOMMENT)
   /**
   //JSSTEP1 EXEC JS6PROC,PRM=S
   //JLRCL DD DISP=SHR,DSN=&DJSNAME..SJJSPARM(&OPT01)
   // DD DISP=SHR,DSN=&DJSNAME..SJJSPARM(&OPT02)
   // DD DISP=SHR,DSN=&DJSNAME..SJJSPARM(&OPT0N)
   // DD DDNAME=SELIN ** CANNOT USE SYSIN HERE **
   //SYSEXEC DD DISP=SHR,DSN=&DJSNAME..SJJJCLST FOR REXX EXEC'S
   //JLSCHO DD SYSOUT=* TO DISPLAY THE JLSCH CA7 STATEMENT
   //JLSCHR DD DISP=(,PASS),SPACE=(TRK,(1,10)),UNIT=&SYSDA FOR JOB/SCAN
   //SYSTSPRT DD SYSOUT=* FOR REXX ERROR & DEBUG MESSAGES
   //SYSOUT DD SYSOUT=* FOR MESSAGES FROM SASSBSTR
   //ERRORS DD SYSOUT=* FOR ERRORS FROM SASSBSTR
   //SYSPRINT DD DISP=(,PASS),SPACE=(TRK,(1,10)),UNIT=&SYSDA FOR SASSBSTR
   //SYSIN DD DISP=(,PASS),SPACE=(TRK,(1,10)),UNIT=&SYSDA FOR SASSBSTR
   //JLCA7LB DD DISP=SHR,DSN=&CA7LIB CA-7 LIBRARY FOR ASG
   //CA7CMDS DD DISP=SHR,DSN=&CA7CMDS CA-7 CMD FILE FOR SASSBSTR
   //UCC7CMDS DD DISP=SHR,DSN=&UCC7CMDS ALT. CA-7 CMD FILE FOR SASSBSTR
9. //BTERM0@I DD DISP=SHR,DSN=&BATCHIN <== CA-7 BATCH INPUT FILE
10. //BTERM0@O DD DISP=SHR,DSN=&BATCHOUT <== CA-7 BATCH OUTPUT FILE
   //JLSCH DD *
11. CA7 SCHID(XXX)
   /** MODIFY THE PRECEDING LINE <== CA-7 PARAMETERS
   /** USE THE FOLLOWING ON CA7 STMT: THE VALUE MUST BE WITHIN ().
   /** *** ALL PARAMETERS EXCEPT CA7 ARE OPTIONAL ***
   /** SCHID- 1 TO 3 BYTES LONG. DEFAULT - 001
   /** SJOB - 1 TO 8 BYTES LONG. DEFAULT - VALIDATION MEMBER NAME
   /** SDATE- 6 BYTES FORMAT YYMMDD. DEFAULT - CURRENT DATE
   /** STIME- 4 BYTES FORMAT HHMM. DEFAULT - CURRENT TIME
   /**
   //SELIN DD * *** THIS IS NORMALLY SYSIN ***
12. INPUT SELECTION
   DSN=YOUR.JCLLIB <== INPUT JCL LIBRARY
   SELECT=YOURJOB <== MEMBER TO SCAN
   SOURCE TYPE=PDS <== TYPE OF INPUT LIB
REPORT OPTIONS
   OPTIONS IN EFFECT =PRINT <== PRINT OPE REPORT?
   STRUCTURED JCL LISTING=PRINT <== PRINT SJL REPORT?
RUN OPTIONS
   SCHEDULER=(CA7,SCAN)

```

## JOB/SCAN CA-7 Interface Setup

This topic describes about the installation of the interface.

### Create a CA-7 Working Library

Create a CA-7 JCL library where the JCA Edit macro, foreground panels, and the batch jobs may write records prior to the LICK request to resolve CA-7 # control statements and CA-Driver procedures. Use either of two methods to make this

library available to CA-7; both require a dataset name and an INDEX value compliant with CA-7 rules. The same dataset name and INDEX are required in [Set Site-Specific Values](#) and in batch job JCL in [Using CA7SCHD to Validate a CA-7 Schedule](#) and in [Using JSCA7 to Validate a JOB](#).

**Method 1:** To define the library immediately and temporarily to CA-7, use the CA-7 JCL command to define the library dynamically. Consult the CA-7 documentation for details about this command. If the CA-7 started task is restarted for any reason, reissue this command.

**Method 2:** Defines the library permanently to CA-7.

### **To define library permanently**

1. Add the following to the CA-7 INIT deck as a valid JCL library:

```
JCL, DSN=XXXX.XXXX.XXXX, INDEX=YYY, DYN=2
```

2. Substitute a dataset name and INDEX for `XXXX.XXXX.XXXX` and `YYY` respectively.
3. Restart the CA-7 started task with the new INIT deck.

### **Set Site-Specific Values**

#### **To set required values for your site, modify member J00CAVAR in the (VB)CLIST library**

1. Replace `XXXX.XXXX` with the name of the JOB/SCAN product high-level qualifier.
2. Set the LIBDEF switch for the JCA Edit macro. Use `YES` to automatically allocate ISPLLIB, ISPPLIB, and ISPMLIB using LIBDEF. Use `NO` to require that each TSO/ISPF session preallocate these libraries before using the JCA Edit macro.
3. Insert CA-7 generic user ID. This might be optional; consult with local system administration.
4. Insert CA-7 password for the ID. If the generic user ID is optional, so is this password.
5. Replace `XXXX.XXXX.XXXX` with the dataset name of the working library created in [Create a CA-7 Working Library](#).
6. Replace `yyy` with the INDEX of the working library created in [Create a CA-7 Working Library](#).
7. Enter the name of the CA7 program based on the interface you use: for CAICCI, use `CAL2X2WB`; for the CA7 Batch Terminal interface, use `SASSBSTR`.
8. Enter the SASSBSTR parameters.
9. Replace `XXXX.XXXX.command` with the actual dataset name for the CA-7 Command file.
10. Replace `CA7CMDS` with the DD name of the CA-7 Command file.
11. Replace `VVVVVVVV` with the volume name for new work files.
12. If the BTI DD is used, replace the @ with a digit from 1–9 in the quoted DD name:  

```
BTERM0@I = 'BTERM0@I'
```
13. Insert the name of the BTI dataset.
14. If the BTO DD is used, replace the @ with a digit from 1–9 in the quoted DD name:  

```
BTERM0@O = 'BTERM0@O'
```
15. Insert the name of the BTO dataset.



## Setting up the JCA Edit Macro

```

/***** REXX for CA-7 JCL validation *****/
/* Member J00CAVAR */
/* ASG JOB/SCAN CA-7 JCL Validation - define variables */
.
.
/*****/
/* TRACE RESULTS */
/*****/
/**** MANUALLY SET THE FOLLOWING VALUES ****/
/*****/
1. djsname = 'XXXX.XXXX' /* JOB/SCAN HIGH LEVEL QUALIFIER */
2. libdef = 'YES' /* USE LIBDEF IN JCA PROCESSING */
3. ca7id = '' /* GENERIC CA-7 USER ID */
4. ca7pd = '' /* PASSWORD FOR CA-7 USER ID */
5. ca7lib = 'XXXX.XXXX.XXXX' /* CA-7 JCL LIBRARY NAME */
6. index = 'YYY' /* CA-7 JCL LIBRARY INDEX */
7. ca7pgm = 'CAL2X2WB' /* USING THIS CA-7 PROGRAM */
/* SPECIFY CAL2X2WB OR SASSBSTR */
8. parm = '' /* PARAMETER PASSED TO SASSBSTR */
9. command = 'XXXX.XXXX.COMMAND' /* CA-7 COMMANDS DATA SET NAME */
10. ucc7cmds = 'CA7CMDS' /* DDNAME FOR CA-7 COMMANDS DD */
11. vol_space = 'VOLUME(VVVVVV) SPACE(3,3) CYL' /*new work files*/
12. BTERM0@I = 'BTERM0@I' /*DD NAME - BATCH TERMINAL INPUT */
13. BTIDSN = 'NULLFILE' /* DATA SET NAME for BTI */
14. BTERM0@O = 'BTERM0@O' /*DD NAME - BATCH TERMINAL OUTPT */
15. BTODSN = 'NULLFILE' /* DATA SET NAME for BTO */
...

```

## Setting up Pooling (Optional)

ASG recommends using pooling so that the Edit macro and batch jobs can use the next available batch terminal datasets in the pool. With pooling disabled, specific batch terminal datasets will have to be hardcoded.

The default name of the procedure is CA7BTI (see the below figure for a sample of this PROC). Consult CA-7 and CA-7 Driver product documentation for the location, use and possible modification of job to establish pooling.

### CA-7 Batch Terminal Interface JCL Procedure (CA7BTI)

```

//CA7BTI PROC ID=0, POOL='1-8', DSNPFX=, OPT=, PG=SASSBSTR
//BTERM EXEC PGM=&PG,
// PARM='&ID, POOL=(&POOL), DSNPFX=&DSNPFX, &OPT'
//STEPLIB DD DISP=SHR, DSN=cai.ca7.loadlib
//UCC7CMDS DD DISP=SHR, DSN=cai.ca7.commds
//SYSPRINT DD SYSOUT=*, DCB=BLKSIZE=133
//ERRORS DD SYSOUT=*, DCB=BLKSIZE=133
//SYSUDUMP DD SYSOUT=*

```

# User Information - JOB/SCAN with ESP

This documentation provides complete information for using the ASG-JOB/SCAN (herein called JOB/SCAN) interface to ESP.

Topic	Description
<a href="#">Getting Started</a>	Outlines major features and benefits of the JOB/SCAN interface to ESP.
<a href="#">Validating Individual Members</a>	Explains the use of the Edit macros specially designed for the ESP interface.
<a href="#">Validating an ESP Event</a>	Describes how to validate the JCL for an ESP event.
<a href="#">Validating a Schedule</a>	Describes how to validate the JCL for an ESP schedule.
<a href="#">Installing ESP</a>	Describes how to install the interface.

## Getting Started

This topic introduces the JOB/SCAN ESP scheduler interface, including terminology, and provides usage examples.

ASG provides this documentation to IT professionals who work with JCL members containing ESP symbols. The crucial point is that you want ESP to resolve the symbols prior to validation by the JOB/SCAN to ESP interface in these situations:

- A single JCL member
- All the jobs for an Event
- An entire day's (or night's) schedule

The facilities provided through the JOB/SCAN ESP interface support this wide variety of requirements and enhance the overall quality and efficiency of the JCL processing at any site using ESP Scheduler (ESP Workload Manager).

This interface supports ESP 5.2 and later.

- [JOB/SCAN-ESP Basic Process](#)
- [Usage Scenarios](#)
- [Terminology](#)

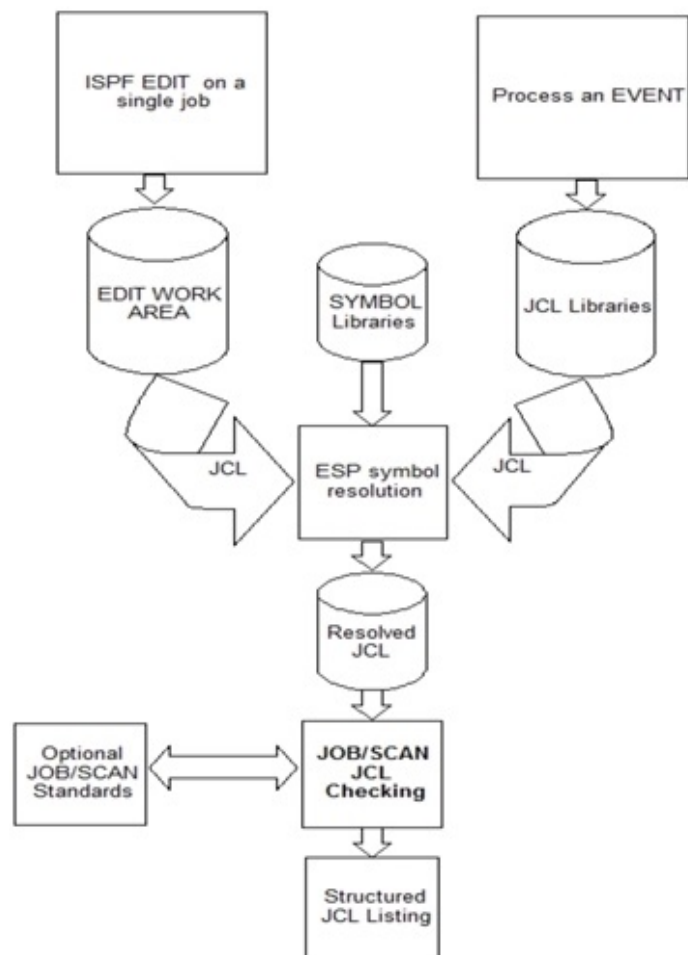
## JOB/SCAN-ESP Basic Process

The basic concept of all the processes described in this documentation is the use of the appropriate symbol library or SYMLIB for the ESP simulation. Because the only way a SYMLIB reference can be made is to place it in an Event definition, it is always necessary to provide an EVENTID whenever a JOB/SCAN-ESP process is invoked. The ESP simulation is invoked

first to perform its resolution based on the appropriate SYMLIB. After resolution is complete, JOB/SCAN is invoked to check the JCL.

The JCL for which resolution is needed comes either from the EDIT session or from specified JCL libraries.

1. ESP Scheduler uses the appropriate Symbol Libraries (SYMLIB) and resolves the ESP symbols.



2. The resolved JCL is placed in a sequential file.
3. JOB/SCAN reads the JCL in the sequential file.
4. Because ESP returns the JCL in execution sequence, as JOB/SCAN performs its validation process, its simulation is all the more complete and accurate.
5. JOB/SCAN invokes a standards program, if requested.
6. JOB/SCAN produces its Structured JCL Listing containing the resolved and expanded JCL with all ESP and JOB/SCAN messages.

### SYMLIBs in JCL Validation for a Single Member

Invoking the process while in ISPF EDIT is somewhat more complex. Because you are editing the JCL outside of its normal inclusion within an Event, the only way to tell ESP which SYMLIB to use is through the ESP Event Criteria panel. The JCL in the EDIT session is passed to ESP along with the specified EVENTID. The resolved JCL is then checked by JOB/SCAN. If you

SYMLIBs, the same JOB/SCAN panel allows you to select easily different EVENTIDs that use different SYMLIBs.

### SYMLIBs in JCL Validation for an Event or Schedule

When validating a specific Event, the Event definition contains the proper SYMLIB reference. When validating a complete schedule, the necessary EVENTIDs are contained in the schedule, so there is no need to specify a particular EVENTID and, consequently, no need to be concerned about SYMLIB references.

JOB/SCAN does an accurate catalog simulation when it validates JCL in the planned execution sequence.

## Usage Scenarios

This interface supports these basic usage scenarios in which resolution of ESP symbols is followed by JOB/SCAN validation:

- You are editing JCL that contains ESP symbols, and you want JOB/SCAN to validate the JCL after the symbols have been resolved.
- You have changed the JCL used by an Event and you want to use JOB/SCAN to validate JCL for the entire Event.
- You are working with Events within the ESP environment and want to validate the JCL used by the Events.
- You want JOB/SCAN to validate an entire batch schedule as ESP will submit it.

To support the first scenario, a JOB/SCAN Edit macro (JESP) passes the necessary parameters to ESP so that ESP uses the appropriate SYMLIB for symbol resolution. The Edit macro then invokes JOB/SCAN to perform validation. The resulting JOB/SCAN Structured JCL Listing (SJL) shows exactly how the resolution was performed and whether JOB/SCAN found any errors in the resolved JCL.

The second scenario is supported by JOB/SCAN either in ISPF foreground or through a batch run. Either way, JOB/SCAN uses its catalog simulation facility to confirm that the changes do not create JCL errors when the Event is next run in production. Because the jobs are prepared in execution sequence, JOB/SCAN's thorough catalog simulation facility is able to check for file availability during the process.

In the third scenario, support is gained through ESP's ISPF interface. In your normal course of working with ESP Events, you can use the JCLSCAN exit on the Simulate Event Execution Additional Options panel ([Validate an Event with the ESP JCLSCAN Exit](#)) to validate JCL for an event. This method integrates the running of ESP's event simulation followed by JOB/SCAN's validation without exiting ESP.

The fourth scenario can be used when you want to check everything just prior to production. It is an effective way to provide an early alert about potential problems so that necessary fixes can be applied prior to production time. A batch job skeleton is provided into which you place the time frame you want to check. The job invokes the ESP extraction and symbol resolution processes. Then, it invokes JOB/SCAN to perform syntax checking and catalog simulation over the entire period selected.

If your site uses a JOB/SCAN exit for standards, it can be invoked in each of these scenarios.

## Terminology

The following terms are used in this documentation:

## Event Simulation

ESP uses the term 'simulation' to refer its process of simulating an event:

- The triggering of an ESP Event.
- The generation of the corresponding JCL.
- The resolution of ESP Symbols in the JCL (see [Symbol Resolution](#)).

## Catalog (Run-Time) Simulation

Thorough validation of production JCL requires that the JCL be analyzed on multiple levels to identify potential problems:

- On the most basic level, JOB/SCAN ensures that the JCL is *syntactically* correct.
- At the *run-time* level, JOB/SCAN identifies:
  - environmental errors, such as security violations and programs not found.
  - dataset disposition errors.

Whether processing a single JCL member or a sequence of members, JOB/SCAN uses its powerful z/OS *catalog simulation* facility to ensure that the JCL steps will create, use and delete the datasets as expected.

## Symbol Resolution

Following are the two types of symbols used in JCL:

- JES Symbols: these symbols, native to the z/OS operating system, are resolved by JES at run time.
- ESP Symbols: ESP substitutes the appropriate value at the time the ESP Event is triggered. It is these symbols which must be resolved by ESP prior to JCL validation by JOB/SCAN.

## Validation

In JOB/SCAN usage, Validation refers to the process of analyzing JCL to identify problems that might occur in production (such as syntax errors, missing resources, security violations, catalog problems, and so on.)

### JCL Checking

This term is synonymous with validation.

## Event

This refers to an ESP Event. In this documentation, the expression 'Validate an ESP Event' refers to the processes by which JOB/SCAN validates the JCL associated with and generated by an ESP Event.



Depictions of some ISPF-based panels in this documentation are truncated to conserve space and remove unnecessary details. Such panels are identified by “...” appearing anywhere in the image of the panel.

# Validating Individual Members

The JOB/SCAN ESP interface can check single members through the Edit macro, through batch or through the JOB/SCAN JCL Validation panel.

- [Validate a Single Member with the JSESP Edit Macro](#)
- [Validate a Single Member in Batch](#)

- [Validate Members from the Validation Panel](#)

## Validate a Single Member with the JSESP Edit Macro

Use the JSESP Edit macro when you are editing a JCL member containing ESP symbols. JSESP invokes symbol resolution, passing the necessary parameters to ESP so that it can perform its symbol resolution. The resulting JCL is then given to JOB/SCAN for validation.

There are two essential considerations for ESP:

- You specify the subsystem running ESP.
- You specify the EVENTID so ESP uses the appropriate SYMLIB reference. Optionally, you can give ESP a ROOTJOB and up to four user parameters.

ESP uses the referenced SYMLIB to perform resolution on the JCL from the EDIT session. After that, JOB/SCAN performs its validation and presents the resulting Structured JCL Listing in BROWSE mode. See [Structured JCL Listing \(SJL\)](#) for more information about the SJL. After review, you can return to the original JCL in the EDIT session for further modification.

If the JCL being checked is used in several Events, you run JOB/SCAN several times, entering each EVENTID in turn into the ESP Event Criteria panel. This checks the JCL with each SYMLIB.



- You must have a personal ESP ID that corresponds to your TSO ID. Consult your ESP administrator for assistance.
- Only those Events to which you have access will be available.
- When using the Edit macro JSESP, the JOB/SCAN run-time options values are derived from JSOPTM, not JSOPT3.

## Using the JSESP Edit Macro Method





Type JESP to start the Resolution/Validation Process

```

EDIT      DSSUSR.ESP.JOBLIB(ESPTST) - 01.23          Columns 00001 00072
Command ==> JESP                                  Scroll ==> PAGE
***** ***** Top of Data *****
000001 //SYSUSR  JOB (JRR),'UNLOAD',MSGCLASS=X,
000002 //        NOTIFY=SYSUSR,MSGLEVEL=(1,1)
000003 //*
000004 //IEFBR14 EXEC PGM=IEFBR14
000005 //DD2     DD  DSN=SYSUSR.JCL.Y%ESPAYEAR,DISP=SHR
    
```

Type in criteria for ESP resolution

```

- JOB/SCAN ----- ESP EVENT CRITERIA -----
COMMAND ==>

PREFIX          ==> SYSUSR          BLANK DEFAULTS TO GROUP NAME
DESCRIPTIVE NAME ==> ESPTST         BLANK LISTS ALL EVENTS

SCHEDULE TIME   ==> TODAY
                (SIMULATED DATE OF EXECUTION, DEFAULTS TO NEXT SCH)

ROOT JOB(S)     ==>
                (NAMES OF JOBS AND SUCCESSORS FOR A PARTIAL RERUN)

USER PARAMETERS TO PASS TO THE EVENT:
USER1 ==>
USER2 ==>
USER3 ==>
USER4 ==>

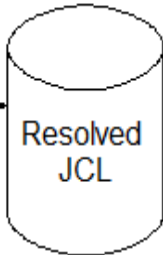
CASE SENSITIVE ==> (BLANK, N OR Y)

ESP SUB SYSTEM ID ==> ESP SPECIFY ESP SUB SYSTEM ID OR BLANK FOR LIST

DISPLAY SIMULATION MESSAGES ==> Y (N OR Y)

PRESS <ENTER> TO LAUNCH ESP SYMBOL RESOLUTION FOLLOWED BY JOB/SCAN VALIDATION.
    
```

ESP:  
1. Resolves the ESP Symbols  
2. Creates a Sequential file



JOB/SCAN:  
1. Validation  
2. Run-time simulation

VIEW  
Structured JCL Listing

<PF3> BACK TO EDIT SESSION  
to REPEAT or SAVE

## Validate a Single Member in Batch

To validate a member in batch (the numbered steps refer to the below figure)

1. Make a copy of the JSMVS member (from the SSJSSAMP library).
2. Change the JOB name and add appropriate fields and parameters to the JOB card.
3. Add the input dataset.
4. Add the member name.
5. Add these DDs after the REPORT OPTIONS statement:

```
//STEP01.SYSPRINT DD DSN=&&SYSPRINT,UNIT=SYSDA,SPACE=(TRK,(1,1)),
// LRECL=133,RECFM=FB
//STEP01.JLSCHO DD SYSOUT=*
//STEP01.JLSCH DD *
```

6. Add this control card directly after the JLSCH DD:

```
ESP EVENT('event.name') SCHEDULE('datespecification') -
ROOTJOB('rootjobname') USER1(PARM) USER2(PARM) USER3(PARM) -
USER4(PARM)
```

For DD JLSCH, these are the fields (if any value contains delimiters or spaces place quotes around it):

- EVENT('event.name') the event name in dotted notation.
- SCHEDULE('datespecification') the date and time of the schedule to be validated.
- ROOTJOB('rootjobnames') - a quoted string containing the names of jobs and successor jobs needed for a partial rerun. (Separate ROOTJOB names with spaces or delimiters.)
- USER1-4('parm') - up to 4 user-supplied parameters.

Precede the first card with ESP. Place a continuation character (-) at the ends of lines that continue to the next line.

### Modifying batch JCL to validate a single member

```
2. //MYESPJOB JOB (ACCT),'PROGRAMMER NAME'
//*****
//* EXECUTE JOB/SCAN (MVS SYSTEMS)
//* REF: SJJSSAMP(JSMVS)
//*
//* NOTE: MODIFY JOB STATEMENT AND ITEMS INDICATED BY ARROWS <==
//*****
//JSSTEP1 EXEC JS6PROC,PRM=S
//SYSIN DD *
INPUT SELECTION
3. DSN=YOUR.JCLLIB <== INPUT JCL LIBRARY
4. SELECT=YOURJOB <== MEMBER TO SCAN
SOURCE TYPE=PDS <== TYPE OF INPUT LIB
REPORT OPTIONS
OPTIONS IN EFFECT =PRINT <== PRINT OPE REPORT?
STRUCTURED JCL LISTING=PRINT <== PRINT SJL REPORT?
//
5. //STEP01.SYSPRINT DD DSN=&&SYSPRINT,UNIT=SYSDA,SPACE=(TRK,(1,1)),
5. // LRECL=133,RECFM=FB
5. //STEP01.JLSCHO DD SYSOUT=*
5. //STEP01.JLSCH DD *
6. ESP EVENT(SYSUSR.ESPSAMP) SCHEDULE('JAN31,2006 8AM')
```

## Validate Members from the Validation Panel

You can validate JCL members containing ESP variables through the JOB/SCAN JCL Validation panel.

1. Start JOB/SCAN. The system displays the the ASG-JOB/SCAN Primary menu.

### Selecting JCL validation

```

----- ASG-JOB/SCAN PRIMARY MENU -----
SELECTION ==> 1

    0 - Hardcopy Output and JOB Statement Definition
    1 - Validate JCL
    2 - Change / Reformat JCL
    3 - Processing Options
    4 - Selection List Maintenance
    5 - Standards Utility Menu
    6 - Assign Alternate Options Members
    ...

USERID - SYSUSR
PREFIX - SYSUSR
DATE - nnnn/02/13
TIME - 10:32
APPLID - JS

```

2. Select Validate JCL.
3. Press Enter.
4. JOB/SCAN displays the JOB/SCAN JCL Validation panel.

### Selecting foreground validation mode

```

----- JOB/SCAN - JCL VALIDATION -----
COMMAND ==> F

    F - Foreground execution mode      P - Processing Options
    B - Batch execution mode           E - Edit JCL member
                                      BR - Browse JCL member

Input Library ==> SYSUSR.JCL
Input Type ==> J (J - JOB JCL, P - PROC, or L - Selection List)
Library Type ==> PDS (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> MYJOB
Exclude Members ==>
First Proclib ==>
Proclib Type ==> (PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
Report Heading ==>
Display ==> Y (Yes/No/ErrorsOnly) Browse/View ==> B (B/V)
Hardcopy ==> N (Yes/No/ErrorsOnly/SpaceOnly/JobErrors)
Options in Effect Listing ==> N (Y/N) SJL PDSOUT ==> N (Y/N)

Selection Exit ==> Parm ==>

```

5. Use the F command to begin processing.
6. JOBSKAN displays the ESP Criteria panel. On this panel you supply the Event name using PREFIX and DESCRIPTIVE NAME fields:

**Selecting a list of Events**

```

----- ESP EVENT CRITERIA -----
COMMAND ==>

PREFIX          ==> SYSUSR          BLANK DEFAULTS TO GROUP NAME
DESCRIPTIVE NAME ==>                BLANK LISTS ALL EVENTS

SCHEDULE TIME   ==>
                (SIMULATED DATE OF EXECUTION, DEFAULTS TO NEXT SCH)
ROOT JOB(S)     ==>
                (NAMES OF JOBS AND SUCCESSORS FOR A PARTIAL RERUN)

USER PARAMETERS TO PASS TO THE EVENT:
USER1 ==>
USER2 ==>
USER3 ==>
USER4 ==>

CASE SENSITIVE ==>                (BLANK, N OR Y)

ESP SUB SYSTEM ID ==>                SPECIFY ESP SUB SYSTEM ID OR BLANK FOR LIST

DISPLAY SIMULATION MESSAGES ==> Y                (N OR Y)

```

7. This example shows the selection of an event from a list.

**Selecting an event**

```

----- SELECT ESP EVENT ----- Row 1 to 2 of 2
COMMAND ==>

        PLACE S NEXT TO EVENT TO SELECT. PRESS END TO PROCESS.

EVENT          NEXT DUE
SYSUSR.ESPTEST REQUIRES TRIGGER
SYSUSR.ESPTEST1 REQUIRES TRIGGER
s SYSUSR.ESPTEST2 REQUIRES TRIGGER
***** Bottom of data *****

```

8. Type s next to the event to use.

9. Press Enter.

10. JOB/SCAN redisplayes the ESP Event Criteria panel. Notice that the PREFIX and DESCRIPTIVE NAME fields are now filled in.

## Validating the member

```

----- ESP EVENT CRITERIA -----
COMMAND ==>

PREFIX          ==> SYSUSR          BLANK DEFAULTS TO GROUP NAME
DESCRIPTIVE NAME ==> ESPTEST2       BLANK LISTS ALL EVENTS

SCHEDULE TIME   ==>
                (SIMULATED DATE OF EXECUTION, DEFAULTS TO NEXT SCH)
ROOT JOB(S)     ==>
                (NAMES OF JOBS AND SUCCESSORS FOR A PARTIAL RERUN)

USER PARAMETERS TO PASS TO THE EVENT:
USER1  ==>
USER2  ==>
USER3  ==>
USER4  ==>

CASE SENSITIVE  ==>                (BLANK, N OR Y)

ESP SUB SYSTEM ID ==>                SPECIFY ESP SUB SYSTEM ID OR BLANK FOR LIST

DISPLAY SIMULATION MESSAGES ==> Y          (N OR Y)

```

11. Fill in other fields if desired.
12. Press Enter. JOB/SCAN displays the messages panel.

## Simulation messages

```

----- ESP SIMULATION MESSAGES ----- Row 1 to 2 of 2
COMMAND ==>

                THE FOLLOWING MESSAGES WERE GENERATED BY THE ESP SIMULATION.
                PRESS ENTER OR END TO CONTINUE.

SIMULATION OF EVENT SYSUSR.ESPTTEST AT 15.25.46 ON THURSDAY JULY 29TH, 2004
JOBS: SYSUSR

```

13. Press Enter. JOB/SCAN processes the JCL and displays the SJL for the members.



When selecting all members of a dataset on the Validation panel, JOB/SCAN processes each member of the dataset returning to the ESP EVENT CRITERIA panel for each one.

## Validating an ESP Event

This topic describes how to validate an ESP event.

There are many situations where you might want to check all the JCL in an Event. Perhaps you have modified some of the execution JCL members, added or changed an ESP symbol, changed the execution sequence, added a new job to the sequence, and so on. Whenever this has happened, get the ESP symbols resolved and then have JOB/SCAN check the entire Event for JCL errors.

Because ESP gives JOB/SCAN the jobs in execution order, JOB/SCAN's catalog simulation becomes all the more complete and accurate complementing its syntax checking. This function can be invoked in a variety of ways:

- [Validate an Event with the TSO JSEVENT Command](#)
- [Validate an ESP Event in Batch](#)
- [Validate an Event with the ESP JCLSCAN Exit](#)

## Validate an Event with the TSO JSEVENT Command

This command can be used anytime you are in ISPF. It is set up during the ESP interface installation. The command invokes the ESP Event Criteria panel as shown in [TSO Command JSEVENT](#) . Fill in the data fields as necessary. After data field checking is complete, ESP resolves symbols first followed by the JOB/SCAN validation. JOB/SCAN displays the resulting Structured JCL Listing in BROWSE mode. It can be printed with the appropriate print command at your location.

**TSO Command JSEVENT**







Type  
TSO JSEVENT  
in any ISPF panel

```

Menu Utilities Compilers Options Status Help
-----
                                ISPF Primary Option Menu
Option ==> TSO JSEVENT
0  Settings      Terminal and user parameters      User ID . : ASGUSR
1  View          Display source data or listings     Time. . . : 08:46
2  Edit          Create or change source data       Terminal. : 3278
3  Utilities     Perform utility functions         Screen. . : 1
4  Foreground   Interactive language processing    Language. : ENGLISH
5  Batch        Submit job for language processing   Appl ID . : ISR
6  Command      Enter TSO or Workstation commands   TSO logon : $MYLOG
7  Dialog Test  Perform dialog testing            TSO prefix: ASGUSR
8  LM Facility  Library administrator functions    System ID : MYSYS1
9  IBM Products IBM program development products MVS acct. : ACCT#
10 SCLM         SW Configuration Library Manager  Release . : ISPF 5.0
11 Workplace   ISPF Workplace (NEW)
D  DB2          Perform DB2 interactive functions
E  EXTRA       ***STROBE, DITTO, FTPB, PDS, TASID ***
M  More         IBM Prods - ISMF SMP HCD SDSF RACF etc
    
```

Specify the Event  
you want to check

```

- JOB/SCAN ----- ESP EVENT CRITERIA -----
COMMAND ==>
PREFIX          ==> ASGUSR          BLANK DEFAULTS TO GROUP NAME
DESCRIPTIVE NAME ==> ESPTST        BLANK LISTS ALL EVENTS

SCHEDULE TIME   ==> TODAY
                (SIMULATED DATE OF EXECUTION, DEFAULTS TO NEXT SCH)

ROOT JOB(S)     ==>
                (NAMES OF JOBS AND SUCCESSORS FOR A PARTIAL RERUN)

USER PARAMETERS TO PASS TO THE EVENT:
USER1 ==>
USER2 ==>
USER3 ==>
USER4 ==>

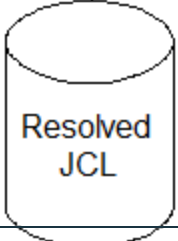
CASE SENSITIVE ==>                (BLANK, N OR Y)

ESP SUB SYSTEM ID ==> ESP          SPECIFY ESP SUB SYSTEM ID OR BLANK FOR LIST

DISPLAY SIMULATION MESSAGES ==> Y          (N OR Y)

PRESS <ENTER> TO LAUNCH ESP SYMBOL RESOLUTION FOLLOWED BY JOB/SCAN VALIDATION.
    
```

ESP:  
1. Collects all JCL for the specified event  
2. Resolves the ESP Symbols  
3. Creates a Sequential file



## Validate an ESP Event in Batch

Another way to have ESP resolve symbols and then use JOB/SCAN to validate an Event is to set up and run a batch job. The template in figure - [Batch Validation JCL Member ESPEVENT](#) is the ESPEVENT member of the JOB/SCAN SJJSSAMP library. You enter the information needed ESP needs to find, collect and prepare the JCL. The sequential dataset is allocated in STEP01. This dataset name is passed to ESP in STEP02. ESP places the resolved JCL into the sequential file. The file is passed to JOB/SCAN in STEP03 for processing and the creation of the Structured JCL Listing. This is an excellent way to verify that changes to the execution JCL will not cause production problems.

The two required input parameters are SIMULATE EVENT and PRINT. If you wish to specify a ROOTJOB or other parameters like USER1, simply add them using proper ESP syntax. For example,

```
SIMULATE EVENT (ASG.SIMJOB) -
  ROOTJOB (MYJOB.A) -
  USER1 (Parameters as needed for ESP) -
  PRINT ('ASG.TEMP.ESPLIST')
```

### *To validate an Event in batch (the numbered steps refer to the below figure)*

1. Make a copy of ESPEVENT.
2. Modify the jobcard.
3. Change the DSN of the temporary dataset on these lines to something workable at your site.
4. Place the ESP subsystem name in the ESP SUB SYSTEM ID field.
5. Verify that the DSN in the STEPLIB for STEP02 is the appropriate name for your installation.
6. Set up the ESP parameters as needed.

After the job is submitted and run, check the resulting SYSPRINT for any error messages from ESP and look in the SJL for JOB/SCAN error messages (the JLKAC DD).



- You might want to make changes to the options settings that JOB/SCAN uses for its processes that are different from the usual ones in use. If you need help with this, consult your JOB/SCAN administrator or call ASG Customer Support.
- Because ESP has already resolved the symbolics in the JCL in STEP02, there is no reason to call the scheduler again in the JOBSCAN step; so, the SCHEDULER run option is set to NONE.
- If you choose to validate multiple jobs through batch, make sure that for each run you use an appropriate event.
- Each job in the Structured JCL Listing contains `//*` comment cards inserted by ESP stating the source library and member name for that particular job. For example:

## Batch Validation JCL Member ESPEVENT

```

2. //ESPEVENT JOB (ACCT), 'PROGRAMMER NAME'

//*****
****
//* SAMPLE BATCH JOB TO VALIDATE EVENT JCL
//* REF: SJJSSAMP (ESPEVENT)

//*****
****
//* MODIFY THIS JCL TO MEET YOUR SYSTEM'S REQUIREMENTS

//*****
****
//***** CREATE DATA SET TO RECEIVE RESOLVED EVENT JCL

//STEP01 EXEC PGM=IEFBR14

3. //DD1 DD DSN=ASG.TEMP.ESPLIST, DISP=(MOD,CATLG), UNIT=SYSDA,
<=====
// SPACE=(TRK, (1,1), RLSE)

//***** EXTRACT THE RESOLVED EVENT JCL
*****
4. //STEP02 EXEC PGM=ESP, PARM='SUBSYS (ESP)', REGION=4M
<=====
5. //STEPLIB DD DSN=CPE.V5R1.SSCPLINK, DISP=SHR
<=====
//SYSPRINT DD SYSOUT=*

//SYSIN DD *

6. SIMULATE EVENT (ASG.SIMJOB) -

3. PRINT ('ASG.TEMP.ESPLIST')

/*

//***** EXECUTE JOB/SCAN
*****
//STEP03 EXEC JS6PROC, PRM=S

//SYSIN DD *

INPUT SELECTION

3. DSN=ASG.TEMP.ESPLIST
<=====
SOURCE TYPE=SEQ

RUN OPTIONS

SCHEDULER=NONE

//***** DELETE THE RESOLVED EVENT JCL
*****
//STEP04 EXEC PGM=IEFBR14

3. //DD1 DD DSN=ASG.TEMP.ESPLIST, DISP=(MOD,DELETE), <=====

```

```
// UNIT=SYSDA, SPACE=(TRK, (1, 1), RLSE)
```

## Validate an Event with the ESP JCLSCAN Exit

There might be times when you are working within ESP to set up or modify an existing Event and you want to check the JCL after symbol resolution. A convenient method is to use JOB/SCAN with the JCLSCAN exit provided within ESP. The procedure to do this for ESP is shown in the below figure.

All the related ESP parameters are taken into consideration by ESP when it does its symbol resolution. A key point is that the prepared JCL from ESP is in execution sequence so the JOB/SCAN validation and catalog simulation will be accurate.



- The exit name to use is `JOBSCAN` not the trademarked product name “JOB/SCAN.”
- The Structured JCL Listing contains a set of `//*` comment cards for each JCL job indicating the library and member name from which the JCL was derived.
- The execution of JOB/SCAN takes place in your TSO address space. You will not be able to do anything else in your address space until the process is finished - usually only a few seconds, but depending on system load and Event size could take a few minutes.
- When the process is done, JOB/SCAN displays the resulting Structured JCL Listing in BROWSE mode.
- You can print it using the appropriate print command.
- Press PF3 to display a list of ESP messages.
- If you put an `*` in the OUTPUT DATASET field on the SIMULATE EVENT EXECUTION panel, ESP makes the resolved JCL available for viewing. To see the resolved JCL, PF3 from the SJL; ESP displays its messages. While viewing the messages press PF7 to page up to the resolved JCL. Press PF3 again to quit and return.



**Validating JCL from ESP Simulate Panel**





CYBESMSM ----- ESP V5.3.0 Main Menu ----- ESP  
 OPTION ==> E

Select an option from the list and press ENTER

Define,Manage,Review...	Functions
E Events	O Set your ESP defaults
A Applications	C Consolidated Status Facility
L Calendars	U ESP Utilities
J Jobs	G ESP Page Mode
T Job Tracking	M ESP Administration
S Scheduled Activity	

Select Events from  
ESP Main Menu

CYBESEVO ----- EVENT MANAGEMENT MENU ----- ESP  
 OPTION ==> 3

Select an option from the list and press ENTER

Selection	Description
1 Quick Event Definition	Quick entry panels to create a simple Event
2 Define a new ESP Event	Full panel series to define a new Event
3 Control an existing Event	Alter, Browse, Delete, Edit, Hold, Release, Simulate, Suspend, Resume or Trigger ESP Events
4 TEST	Test schedule criteria

Select Control an  
existing Event

CYBESECM ----- EVENT CONTROL MENU ----- ESP  
 OPTION ==>

A Alter Event attributes	M Simulate Event execution
B Browse one or more Events	P Release Event(s) for processing
D Delete one or more Events	R Resume Event processing
E Edit Event definition(s)	T Trigger Event execution
H Hold Event processing	U Suspend Event processing
blank Display a List of Events	

DO YOU WANT TO CONFIRM A DELETE REQUEST? ==> Y (Y/N) (Blank defaults to YES)

SINGLE EVENT To process one Event, complete the following.

PREFIX ==> ASGUSR (Blank defaults to group name)

DESCRIPTIVE NAME ==> (Required for single Event)

Enter PREFIX  
and leave  
DESCRIPTIVE  
NAME blank;  
Press <Enter>

CYBESLST ----- LIST EVENTS ----- Row 1 to 8 of 8  
 COMMAND ==>

ENTER ANY COMBINATION OF THESE CODES NEXT TO THE EVENT(S) YOU WANT TO PROCESS

A Alter	D Delete	H Hold	R Resume	M Simulate
B Browse	E Edit	P Release	U Suspend	T Trigger

EVENT	NEXT DUE	SYS	HC	SC
ASGUSR.DSNCREAT	REQUIRES TRIGGER	ASG1	0	0
ASGUSR.MYTEST1	16.15.00 ON THU JUL 1ST, 2004	ASG1	0	0
m ASGUSR.NEWGREETING	15.15.00 ON THU JUL 1ST, 2004	ASG1	0	0
ASGUSR.PROJCLE	REQUIRES TRIGGER	ASG1	0	0
ASGUSR.TESTNUMBER1	15.00.00 ON THU JUL 1ST, 2004	ASG1	0	0
**END**				

Select an EventID with  
M for "Simulate" from  
the list;  
Press <Enter>

CYBESSIM ----- SIMULATE EVENT EXECUTION ----- ESP  
 COMMAND ==>  
 OUTPUT DATASET ==>

## Validating a Schedule

This topic describes how to validate a schedule.

There might be times when you want to perform a JOB/SCAN validation on an entire night's production batch run. For this batch-oriented task, you use the existing JOB/SCAN SJJSSAMP JCL member, ESPSCHED. The batch job executes these steps:

- STEP01 allocates the temporary dataset into which ESP places the resolved JCL.
- STEP02 runs ESP to prepare the JCL.
- The JOBSCAN step invokes JOB/SCAN for validation of the resolved JCL in production sequence.

The temporary dataset is referenced as ddname JCLFILE in its allocation in STEP01, referenced as JCLOUT in the SYSIN parameters for ESP in STEP02 and as ddname JLJOB for use by JOB/SCAN in STEP03. Be sure the same dataset name is used in all three places. If you want to look at the resolved JCL for troubleshooting purposes, change the DISP in ddname JLJOB to KEEP.

In brief, when the run is complete, you first check the ESPSCHED job output for return codes to make sure everything ran as planned. Then, look for any errors that JOB/SCAN has found that can cause problems during the batch cycle. Accordingly, review the error messages in the Structured JCL Listing (the JLKAC DD) and determine what needs to be done to make the corrections prior to the start of the shift.

**To validate a schedule (the numbered steps apply to the below figure).**

1. Copy the ESPSCHED member from the JOB/SCAN SJJSSAMP library.
2. Modify the job card.
3. Change the DSN of the temporary dataset to something workable at your site. It is referenced in these places.
4. On the SADGEN control statement, specify the starting and ending dates and times.
5. Substitute a valid dataset name for the job to use in the JCLFILE and JLJOB DD statements and in the SADGEN input cards (as shipped the dataset name is `TEMP.SCHEDULE.JCLOUT`).



- You might want to make changes to the options settings that JOB/SCAN uses for its processes that are different from the usual ones in use. If you need help with this, consult your JOB/SCAN administrator or call ASG Customer Support.
- Because ESP has already resolved the symbolics in the JCL in STEP02, there is no reason to call the scheduler again in the JOBSCAN step; so, the SCHEDULER run option is set to NONE.
- Each job in the Structured JCL Listing will have `/**` comment cards inserted by ESP stating the source library and member name for that particular job. See [Reference Information](#) for more information about the SJL.

### Validating a Schedule in Run-time Order

```

2. //ESPSCHD1 JOB (ACCT),'PROGRAMMER NAME'
   //*****
   /*  INTERFACE TO ESP (ENTERPRISE SYSTEMS PLATFORM)
   /*  REF:  SJJSSAMP(ESPSCHED)
   //*****
   //***** EXTRACT THE JOBS FROM ESP *****
   //STEP01 EXEC PGM=IEFBR14
3. //JCLFILE DD UNIT=SYSDA,SPACE=(CYL,(5,5)),DISP=(NEW,CATLG),
5. //      DSN=TEMP.SCHEDULE.JCLOUT
   //STEP02 EXEC ESP,PARM='SAR'
   //SYSPRINT DD SYSOUT=*
   //SYSIN DD *
4. SADGEN DATASET(SADFILE) -
   FROM('00:01 TODAY') TO('24:00 TODAY') -
5.      JCLOUT('TEMP.SCHEDULE.JCLOUT')
   /*
   //***** EXECUTE JOB/SCAN *****
   //JOBSCAN EXEC JS6PROC,PRM=S,
   //      COND=(8,LT)
5. //JLJOB DD DSN=TEMP.SCHEDULE.JCLOUT,DISP=(OLD,DELETE)
   //SYSIN DD *
   INPUT SELECTION DO NOT REMOVE
   SOURCE TYPE=SEQUENTIAL DO NOT REMOVE
   RUN OPTIONS DO NOT REMOVE
   SCHEDULER=NONE DO NOT REMOVE
   //

```

## Installing ESP

This topic describes the requirements and procedure for installing the ESP interface.

### Version Support for New JOB/SCAN Customers

Starting with version 6.2.8A, enhanced support for the ESP Scheduler is fully integrated into JOB/SCAN. It is not necessary to install additional support (from the JESP FunctionPack) to enhance the ESP scheduler interface. Follow the instructions in this section to install the interface.

### Version Support for Existing JOB/SCAN Customers

Starting with version 6.2.8A, enhanced support for the ESP Scheduler is fully integrated into JOB/SCAN. It is no longer necessary to concatenate the JESP FunctionPack data sets ahead of JOB/SCAN product data sets. If you previously installed the FunctionPack, first remove these concatenations prior to installing the ESP Scheduler Interface and rename or archive the FunctionPack data sets. Then, continue with the instructions in this section to install the interface.

- [Modify the CLIST Library \(SJSCLST\) Members](#)
- [Activate the ESP Scheduler Interface](#)
- [Verify Installation](#)

## Modify the CLIST Library (SJJSCCLST) Members

### To make these modifications to members in the SJJSCCLST library

1. Modify the JSESP EXEC in the SJJSCCLST library. Change the string for the `djsname` variable from `xxxxx.xxxxx` to the JOB/SCAN data set qualifier suited for your site.

```
djsname = 'XXXXX.XXXXX' /* Change to your JOB/SCAN dsn qualifier */
```

2. Modify the J00CEAL EXEC in the SJJSCCLST library. Change the string for the `djsname` variable from `xxxxx.xxxxx` to the JOB/SCAN data set qualifier suited for your site.

```
djsname = 'XXXXX.XXXXX' /* Change to your JOB/SCAN dsn qualifier */
```

3. Modify the J00CEVNT EXEC in the SJJSCCLST library. Change the string for the `djsname` variable from `xxxxx.xxxxx` to the JOB/SCAN data set qualifier suited for your site.

```
djsname = 'XXXXX.XXXXX' /* Change to your JOB/SCAN dsn qualifier */
```



- If you used LIBDEFs for the original installation of the JOB/SCAN Edit macros, include the LIBDEFs in the JSESP and J00CEVNT EXECs. Refer to the EXECs for placement of these definitions. Example (address ISPEXEC):

```
"LIBDEF ISPPLIB DATASET ID('"DJSNAME".SJJSPLIB') "  
"LIBDEF ISPMLIB DATASET ID('"DJSNAME".SJJSMLIB') "  
"LIBDEF ISPLLIB DATASET ID('"DJSNAME".SJJSLMOD') "  
"LIBDEF ISPSLIB DATASET ID('"DJSNAME".SJJSSLIB') "
```

- The JOB/SCAN load library, SJJSLMOD, must be allocated to the ISPLLIB DD, or it must be accessible from the LINKLIST.
- Refer to [Installation](#) for more about setting up JOB/SCAN libraries.

## Activate the ESP Scheduler Interface

Do all the parts of this section from [Set the SCHEDULER Run Option](#) to [Background Processing from JOB/SCAN Panels](#), inclusive.

### Set the SCHEDULER Run Option

Set the SCHEDULER Run Option to ESP in the JSOPT02 member in the SJJSPARM library:

```
RUN OPTIONS  
...  
SCHEDULER=(ESP, subsystemname)
```

where `subsystemname` is a one- to four-character string naming the ESP subsystem.

### Turn on the Scheduler Processing Option

1. Choose Processing Options on the JOB/SCAN JCL Validation panel. (Alternately, select processing options from the ASG-JOB/SCAN Primary Menu, if available.)

#### Choosing Processing Options from Validation panel

```

----- JOB/SCAN - JCL VALIDATION -----
COMMAND ==> P

      F - Foreground execution mode          P - Processing Options
      B - Batch execution mode              E - Edit JCL member
                                          BR - Browse JCL member

Input Library  ==> SYSUSR.JCL
Input Type    ==> J      (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ==> PDS    (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> BR14
Exclude Members ==>
First Proclib ==>
Proclib Type  ==>      (PDS, PAN, LIB, or LIB-3)
...
    
```

2. Type a Y in the Use SCHEDULER interface field.
3. Press PF3 to return and accept this change.

#### Processing Options Panel

```

D02PEP ----- JOB/SCAN PROCESSING OPTIONS -----
COMMAND ==>

Use Error Table          ==> N      (Y/N)
Condition Code Processing ==> N      (Y/N)
Catalog Access          ==> Y      (Y/N)
VTOC Access             ==> Y      (Y/N)
PDS Member Verification ==> Y      (Y/N)
Security Exit DSNNAME Check ==> Y    (Y/N)
Program Checking        ==> Y      (Y/N)
Disable SMS Processing   ==> N      (Y/N)
IDCAMS Processing       ==> Y      (Y/N)
Use SCHEDULER interface ==> Y      (Y/N)
Job Limit               ==>        (1 to 999 or Blank For No Limit)
Access Control Cards     ==> Y      (Y/N)
...
    
```

### Activate DDs in FILE ALLOCATIONS Statement

Make sure that the attributes for the JLSCHR DD in the FILE ALLOCATIONS statement in JSOPT02 are suitable for your site.

```

JLSCHR =DSN=&&JLSCHR,NEW,DELETE,          SCHED WORK FILE
        UNIT=SYSDA, VOLSER=OPTIONAL,
        TRKS, PRIM=2, SEC=1, RECFM=FB,
        LRECL=80, BLKSIZE=27920
    
```

### Control Table CE: Scheduler Restart Program Names

Control Table CE contains the names of scheduler restart programs. JOB/SCAN simulates the operation of the restart program when one of these programs is found. Control Table CE entries are required if the customer site is using a scheduling product.

#### Format

Enter Control Table CE data as follows:

Column	Entry
6-7	CE
8-15	Scheduler restart program name

### Example

```
.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....
      CEU11RMS
      CECTRCTR
      CEEQQDELDS
```

### Add DD statements to batch jobs

If you intend to run the JOB/SCAN ESP interface in batch, add these DD statements to the JCL running JS6PROC:

```
//STEP01.SYSPRINT DD DSN=&&SYSPRINT,UNIT=SYSDA,SPACE=(TRK,(1,1)),
// LRECL=133,RECFM=FB
//STEP01.JLSCHO DD SYSOUT=*
//STEP01.JLSCH DD *
```

For DD JLSCH, this table lists the control card values (if any value contains delimiters or spaces, surround it with quotes):

Control Card Value	Description
EVENT(eventname)	The event name in dotted notation.
SCHEDULE(datespecification)	The date and time of the schedule to be validated.
ROOTJOB(rootjobnames)	The names of jobs and successor jobs needed for a partial rerun.
USER1-4	Up to 4 user-supplied parameters of the form, USER1(YOUR TEXT HERE).

Precede the first card with the characters ESP. Place a continuation character (-) at the ends of lines that continue to the next line. For example:

```
ESP ROOTJOB(MYJOB.A) -
  USER1(Parameters as needed for ESP)
```

### Background Processing from JOB/SCAN Panels

If you intend to run the JOB/SCAN ESP interface in background mode from JOB/SCAN panels, add these DD statements directly to JS6PROC:

```
//SYSPRINT DD DSN=&&SYSPRINT,UNIT=SYSDA,SPACE=(TRK,(1,1)),
// LRECL=133,RECFM=FB
//JLSCHO DD SYSOUT=*
//JLSCH DD DSN=MY.CNTL.CARD,DISP=SHRS
```

The JLSCH DD should refer to a data set containing the control cards as described in [Add DD statements to batch jobs](#).

## Verify Installation

Use the procedures in this section to verify the installation of the interface. You need a userID defined to TSO and an identical personal ID defined to ESP.

## Verify the Edit Macro JSESP

### *To confirm that the process works for the Edit macro JSESP*

1. Edit a member containing ESP symbols.
2. Type the command `JSESP`.
3. Press Enter.
4. Confirm that JOB/SCAN displays the ESP Event Criteria panel.
5. Press Enter.
6. JOB/SCAN should display a list of subsystem IDs.
7. Select with `s` the one that is running ESP.
8. Press Enter. JOB/SCAN returns to the ESP Event Criteria panel.
9. Press Enter.
10. JOB/SCAN should display a list of EVENTIDs.
11. Select an EVENTID for this test.
12. Press Enter. JOB/SCAN displays the ESP Messages panel if you entered `Y` in the DISPLAY SIMULATION MESSAGES field.
13. Press Enter. JOB/SCAN finishes by displaying the SJL with the symbols resolved.
14. Check the ESP and JOB/SCAN messages.

Each user goes through these steps the first time they run JOB/SCAN to resolve ESP symbols.

## Verify Checking from the ASG-JOB/SCAN Primary Menu

### *To confirm the Member checking from the JOB/SCAN JCL Validation panel*

1. Start JOB/SCAN product.
2. On the ASG-JOB/SCAN Primary Menu, choose the Validation option.
3. Fill out the Input Library and Select member fields.
4. Type the `F` command on the command line.
5. Press Enter. Confirm that JOB/SCAN displays the ESP Event Criteria panel. If you have performed the previous verification, this panel contains values.
6. Put the name of a subsystem running ESP in the ESP SUB SYSTEM field.
7. Put in an Event ID. JOB/SCAN displays the ESP Messages panel if you entered `Y` in the DISPLAY SIMULATION MESSAGES field.
8. Press Enter.
9. JOB/SCAN should finish by displaying the SJL with the symbols resolved.
10. Check the ESP and JOB/SCAN messages.

## Verify Event Checking with JCLSCAN Exit

### **To confirm the JCLSCAN exit from ESP**

Follow the instructions given in [Validate an Event with the ESP JCLSCAN Exit](#) for details about this step.

1. Start the ESP product interface.
2. Under Define, Manage, Review... choose Events.
3. Choose Control an Existing Event.
4. Choose Simulate. Whether you make entries into the Single Event fields is not important. If you don't, ESP displays a list.
5. On the Simulate Event Execution panel, type a Y in the Additional Options field.
6. In the JCLSCAN Exit field on the Additional Options panel, type JOBSCAN.
7. Press Enter.
8. JOB/SCAN finishes by displaying the SJL with the symbols resolved.
9. Press PF3 in the SJL to see the ESP messages.
10. Check the ESP and JOB/SCAN messages.

## Verify the TSO JSEVENT command

### **To confirm the availability of the JSEVENT command**

Follow the instructions given in [Validate an Event with the TSO JSEVENT Command](#) for details about this step.

1. In any ISPF panel, type the command `TSO JSEVENT`.
2. Confirm that JOB/SCAN displays the ESP Event Criteria panel. This panel might have values in the fields, depending upon previous verification steps accomplished.
3. Adjust the fields as necessary.
4. Press Enter. JOB/SCAN displays the ESP Messages panel if you entered Y in the DISPLAY SIMULATION MESSAGES field.
5. Press Enter.
6. JOB/SCAN should finish by displaying the SJL with the symbols resolved.
7. Check the ESP and JOB/SCAN messages.

## Verify Single Job Validation in Batch

### **To confirm the single job validation in batch**

1. Follow the instructions in [Validate a Single Member in Batch](#) to verify single job validation through batch.
2. Verify that in the SJL produced the ESP variables are resolved.

## Verify Batch Setup

### **To confirm the batch job ESPEVENT**

Use the instructions in [Validate an ESP Event in Batch](#) for details.



1. In ISPF EDIT, make these modifications to the ESPEVENT JCL:
  - Modify the jobcard.
  - Make the changes to the member to values suitable to your site at the places marked with <====.
  - Place the ESP subsystem name in the ESP SUB SYSTEM ID field.
  - Set up the ESP parameters as needed.

### Modifying the ESPEVENT JCL

```
//ESPEVENT JOB (ACCT), 'PROGRAMMER NAME'
//*****
//* SAMPLE BATCH JOB TO VALIDATE EVENT JCL
//* REF: SJJSSAMP (ESPEVENT)
//*****
//* MODIFY THIS JCL TO MEET YOUR SYSTEM'S REQUIREMENTS
//*****
//***** CREATE DATA SET TO RECEIVE RESOLVED EVENT JCL
//STEP01 EXEC PGM=IEFBR14
//DD1 DD DSN=DSSI.TEMP.ESPLIST, DISP=(MOD,CATLG), UNIT=SYSDA, <====
// SPACE=(TRK, (1,1), RLSE)
//***** EXTRACT THE RESOLVED EVENT JCL *****
//STEP02 EXEC PGM=ESP, PARM='SUBSYS (ESP)', REGION=4M <====
//STEPLIB DD DSN=CPE.V5R1.SSCPLINK, DISP=SHR <====
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
SIMULATE EVENT (DSSI.SIMJOB) -
PRINT ('DSSI.TEMP.ESPLIST')
/*
//***** EXECUTE JOB/SCAN *****
//STEP03 EXEC JS6PROC, PRM=S
//SYSIN DD *
INPUT SELECTION
DSN=DSSI.TEMP.ESPLIST <====
SOURCE TYPE=SEQ
RUN OPTIONS
SCHEDULER=NONE
//***** DELETE THE RESOLVED EVENT JCL *****
//STEP04 EXEC PGM=IEFBR14
//DD1 DD DSN=DSSI.TEMP.ESPLIST, DISP=(MOD,DELETE), <====
// UNIT=SYSDA, SPACE=(TRK, (1,1), RLSE)
```

2. Save the changes.
3. Submit the job.
4. When the system returns a job finished message, go to the SDSF facility or similar to confirm that the job ran and produced the SJL in this data set JLKAC.

### To confirm the batch job ESPSCHEd

1. Use the instructions in [Validating a Schedule](#) for details.
2. In ISPF EDIT, make these modifications to the ESPSCHEd JCL;
  - Modify the job card.
  - Change the DSN of the temporary data set to something workable at your site. It is referenced in these places.
  - On the SADGEN control statement, specify the starting and ending dates and times.

- Substitute a valid data set name for the job to use in the JCLFILE and JLJOB DD statements and in the SADGEN input cards (as shipped the data set name is `TEMP . SCHEDULE . JCLOUT`).
3. Save the changes.
  4. Submit the job.
  5. When the system returns a job finished message, go to the SDSF facility or similar to confirm that the job ran and produced the SJL in this data set JLKAC.

# User Information - JOB/SCAN with TWS

This documentation describes the interface between ASG-JOB/SCAN (herein called JOB/SCAN) and the TWS Scheduler. It explains how to initialize and execute the interface, and provides Edit macro and batch mode instructions.

Topic	Description
<a href="#">Getting Started</a>	Describes major features and benefits of the JOB/SCAN interface to TWS.
<a href="#">JOB/SCAN Commands for TWS</a>	Explains how JOB/SCAN interacts with TWS.
<a href="#">Validate JCL for a Single Member</a>	Describes the use of JOB/SCAN through an Edit macro.
<a href="#">Validating a Plan</a>	Describes how to validate the JCL for a TWS plan.
<a href="#">Validate Application JCL</a>	Describes how to validate the JCL for a TWS plan.
<a href="#">Installation</a>	Describes the installation of the interface.
<a href="#">JCL Parameters for the JVAL Command JCL</a>	Lists the parameters for the JCL generated by the panels launched by the JVAL command.
<a href="#">Preserving Access to an Existing JCL Edit Tool</a>	Explains how to preserve and provide access to a previously installed JCL Edit tool.

## Getting Started

This documentation is designed for a diverse audience: from those who develop, modify or manage test JCL, to those who carefully manage and manipulate production JCL contained within applications, plans and schedules. The features described in this documentation can help technical personnel manage JCL members as they move the JCL from test into production.

- [JOB/SCAN TWS Interface Overview](#)
- [JOB/SCAN TWS Interface Basic Usage Concepts](#)

## JOB/SCAN TWS Interface Overview

JOB/SCAN provides a variety of JCL processing functions, from simple syntax checking, to complex application simulation, to extensive standards checking and enforcement. For sites where the Tivoli Workload Scheduler for z/OS (TWS) from IBM is the tool of choice in managing daily production job runs, ASG offers an advanced interface specifically designed to help IT professionals manage JCL containing TWS JCL variables and job-tailoring directives.

With this interface, JOB/SCAN can process:

- A single job.
- A TWS application.
- The Current Plan.
- Trial Plans on a controller system.

From there, JOB/SCAN:

- Sequences the jobs according to TWS dependencies.
- Resolves variables.
- Expands PROCs and control cards for key utilities.
- Simulates the execution of the JCL to form a complete and accurate run-time picture.

Employ the TWS interface in one of several different ways depending upon the activities performed.

- While working on JCL contained in a TWS PDS library or your own private work library a powerful ASG EDIT macro can be used in VIEW and EDIT modes to resolve variables and directives and to validate the JCL.
- As a scheduling professional, you might have requirements for the:
  - daily validation of jobs in the current plan.
  - advanced validation of a portion of the current plan.
  - validation of a potential trial plan.
  - resolution of JCL errors in a failed job.

All of these tasks can now be performed with JOB/SCAN from within the TWS ISPF product environment.

The interface supports the capability of resolving variables and validating JCL when the TWS Controller and JOB/SCAN are on different LPARs. The interface also supports multiple TWS Controllers on a single LPAR.

## JOB/SCAN TWS Interface Basic Usage Concepts

This documentation employs user scenarios to describe various ways IT professionals can utilize this interface to accomplish critical tasks to ensure an error-free production environment. It is important that users of this interface be familiar with JCL and the TWS facilities and concepts.

### Application Validation

#### *To use the TWS interface to validate a TWS application*

1. Select one of these options:
  - Invoke the JVAL primary command from the TWS Browsing an Application panel or the TWS Editing an Application panel.

**Or**

  - Select Application from the JOB/SCAN FOR TWS -- JCL VALIDATION panel.
2. JOB/SCAN builds and submits job JCL to validate the Application JCL.
3. User inspects the job output.
4. User returns to initial ISPF environment.

## Single member Validation

### *To use the TWS interface on a single JCL member*

1. User supplies basic information including, the TWS subsystem name, the Input Arrival time, the application name, the operation number, and the variable table name to JOB/SCAN through the panel or directly into the batch member
2. TWS resolves any JCL variables.
3. JOB/SCAN performs its validation process on the resolved JCL.
4. JOB/SCAN produces the Structured JCL Listing containing the resolved JCL, TWS processing comments and JOB/SCAN error messages, if any.
5. User inspects Structured JCL Listing.
6. User returns to ISPF Edit session.

## Plan Validation

### *To use the TWS interface to validate a plan (Current Plan or Trial Plan)*

1. Invoke the JOB/SCAN FOR TWS -- JCL VALIDATION panel using one of these options:
  - The JVAL TSO command.

**Or**

  - A user function from the ASG-JOB/SCAN Primary Menu.
2. User chooses the plan.
3. JOB/SCAN builds and submits job JCL to validate plan JCL.
4. User inspects the job output.
5. User returns to JOB/SCAN FOR TWS -- JCL VALIDATION panel.

## Terminology

The following terms are used in this documentation:

### **Catalog (Run-Time) Simulation**

Thorough validation of production JCL requires that the JCL be analyzed on multiple levels to identify potential problems:

- On the most basic level, JOB/SCAN ensures that the JCL is *syntactically* correct.
- At the *run-time* level, JOB/SCAN identifies:
  - environmental errors, such as security violations and programs not found.
  - data set disposition errors.

Whether processing a single JCL member or a sequence of members, JOB/SCAN uses its powerful *z/OS catalog simulation* facility to ensure that the JCL steps will create, use and delete the data sets as expected.

### **Symbol Resolution**

For this documentation, there are two main types of symbols used in JCL:

- Those native to the z/OS operating system and resolved by JES at run time.
- TWS JCL Variables. TWS substitutes the appropriate values at the time the resolution is requested. It is these symbols which must be resolved by TWS prior to JCL validation by JOB/SCAN.

TWS job-tailoring directives are also resolved by the JOB/SCAN-TWS interface.

### Validation(also known as JCL Checking)

In JOB/SCAN usage, *validation* refers to the process of analyzing JCL to identify possible problems before it is put into production (such as, syntax errors, missing resources, security violations, catalog problems, and so on).



Depictions of some ISPF-based panels in this documentation are truncated to conserve space and remove unnecessary details. Such panels are identified by “...” appearing anywhere in the image of the panel.

## JOB/SCAN Commands for TWS

This topic explains how JOB/SCAN interacts with TWS interface.

- [JVAL Command](#)
- [From the ASG-JOB/SCAN Primary Menu](#)

## JVAL Command

Use the JVAL Command from any TWS ISPF panel. There are two contexts for the JVAL command: Generic and Specific. For the JVAL command, all the parameters are optional and mutually exclusive.

Parameter	Description
MENU	JOB/SCAN displays the JOB/SCAN for TWS top menu. This is the default action.
APPL or 1	JOB/SCAN displays the JOB/SCAN for TWS Application Validation panel.
CP or 2	JOB/SCAN displays the JOB/SCAN for TWS Current Plan Validation panel.
TP or 3	JOB/SCAN displays the JOB/SCAN for TWS Trial Plan Validation panel.



Using the JVAL command with incorrect or multiple parameters, JOB/SCAN defaults to the MENU parameter.

### EXAMPLES:

```
JVAL
JVAL CP
JVAL 1
```

## From the ASG-JOB/SCAN Primary Menu

If installed, the **S** option from the ASG-JOB/SCAN Primary Menu invokes the JOB/SCAN for TWS JCL VALIDATION menu.



This command name is set by the JOB/SCAN installer and might be different from 'S' used throughout this documentation. Also, the content of this menu is set by the PARMLIB members in force for a given user. See [System Administrator Tasks](#) for details about setting up PARMLIB members.

### Starting the JOB/SCAN for TWS interface from the JOB/SCAN main menu

```

----- ASG-JOB/SCAN PRIMARY MENU -----
SELECTION ==> S

    0 - Hardcopy Output and JOB Statement Definition
    1 - Validate JCL
    2 - Change / Reformat JCL
    3 - Processing Options
    4 - Selection List Maintenance
    5 - Assign Alternate Options Members
    6 - Product Administration Menu
    7 - JOB/SCAN Usage Report
    S - Validate TWS for z/OS Scheduler JCL
    X - Exit

USERID - SYSUSR
PREFIX - SYSUSR
DATE - 04/11/2005
TIME - 09:02
APPLID - JS

```

Use the usual ISPF/TSO command fast-path to bypass the ASG-JOB/SCAN Primary Menu and go straight to the JOB/SCAN for TWS submenu of choice. For example, `S;2` goes directly to JOB/SCAN FOR TWS - CURRENT PLAN VALIDATION panel.

### Fast-path to a JOB/SCAN for TWS submenu

```

----- ASG-JOB/SCAN PRIMARY MENU -----
SELECTION ==> S;2

    0 - Hardcopy Output and JOB Statement Definition
    1 - Validate JCL
    2 - Change / Reformat JCL
    3 - Processing Options
    4 - Selection List Maintenance
    5 - Assign Alternate Options Members
    6 - Product Administration Menu
    7 - JOB/SCAN Usage Report
    S - Validate TWS for z/OS Scheduler JCL
    X - Exit

USERID - SYSUSR
PREFIX - SYSUSR
DATE - 04/11/2005
TIME - 09:02
APPLID - JS

```

## Validate JCL for a Single Member

This topic describes the use of JOB/SCAN through an Edit macro.

- [JTWS Edit Macro Syntax and Parameters](#)
- [Using the JTWS EDIT Macro in ISPF Edit or View](#)
- [Using the JTWS EDIT Macro from TWS Panels](#)

- Validate Members from the Validation Panel
- Validate a Single Member in Batch

## JTWS Edit Macro Syntax and Parameters

The purpose of the Edit macro is to validate a single member. JTWS uses these command-line optional parameters:

Parameter	Description	Abbr.
COND(jobname. jobstep.stepname. returncode)	Simulates a return code for the specified step. Use asterisks in place of the values not needed. (For more information, see - <a href="#">Control Table AR: Return Code Simulation</a> for the descriptions of Control Table AR and the Processing Specifications Run Option.)	C
DISPLAY(B) or DISPLAY(V)	Displays the Structured JCL Listing (SJL) in Browse mode (B) or View mode (V). The display mode is stored in an ISPF profile variable and is used on subsequent JTWS invocations. The default display is Browse.	D
STDPGM (xxxxxxxx)	Provides the name of the standards program containing site-specific standards to be validated. This parameter may be overridden by an Access Control statement. (See <a href="#">Selection Exit Interface</a> )	ST
PARM(x . . . x)	Specifies a parameter, if any, to be used by the standards program.	P
OPT03 (membername)	Specifies an alternate SJJSPARM library member to be used instead of JSOPT03. This parameter may be overridden by an Access Control statement.	NONE
OPTEM (membername)	Specifies an alternate SJJSPARM library member to be used instead of JSOPTEM.	NONE
RESET	Causes JOB/SCAN to display the advisory panels.	NONE
SELEXIT (xxxxxxxx)	Provides the name of a selection exit program to preprocess the JCL before JOB/SCAN validates it. (See <a href="#">Selection Exit Interface</a> )	SE
SELPARAM(x . . . x)	Specifies a parameter, if any, to be used by the selection exit program.	SP

## Using the JTWS EDIT Macro in ISPF Edit or View

To use this method, you Edit/View a member in an ISPF EDIT/VIEW session. To achieve the most accurate validation of JCL members with JOB/SCAN, the TWS variables must be resolved before JCL validation. The JTWS Edit macro first resolves JCL variables then validates the JCL.



**To validate JCL from within an EDIT/VIEW session**

1. Bring the JCL member into ISPF EDIT.
2. Type JTWS and command line parameters on the command line.
3. Press Enter.

**Edit session showing JTWS command usage**

```

EDIT          SYSUSR.TEST.JCL(BR14JOB) - 01.05          Columns 00001 00072
Command ==> JTWS          Scroll ==> PAGE
***** ***** Top of Data *****
000100 //BR14JOBA JOB
000200 //*%OPC SCAN
000300 //*%OPC TABLE NAME=USR321#VARS
000400 /* OADID DEPENDENT VARIABLE (MODNAME)          = &MODNAME
000500 /* OCCURRENCE APPLICATION ID (OADID)           = &OADID
000600 /* OCCURRENCE OWNER (OADOWNER)                 = &OADOWNER
000700 /* OCCURRENCE IA DATE TIME (OYMD1 OHMM)       = &OYMD1 &OHMM
000800 /* OPERATION JOB NAME (OJOBNAME)              = &OJOBNAME
...

```

4. JOB/SCAN displays this advisory panel to let you know that the TWS JS FILE will be temporarily modified. This change is removed at the end of the session.

**Continuing processing with JOB/SCAN FOR TWS ADVISORY popup**

```

----- JOB/SCAN FOR TWS ADVISORY -----
COMMAND ==> Y

Y - to continue
N - to return

JTWS updates the occurrence JCL in the JS FILE. After
TWS resolves the variables, JTWS restores this Job
to its original contents.

X Enter "X" to suppress this panel in the future.

```



You may opt to disable the reappearance this advisory panel by typing **x** in the Enter "X" to suppress the panel in the future field. Type **Y** on the command line and press Enter.

5. JOB/SCAN displays the JOB/SCAN FOR TWS OPERATION SELECTION CRITERIA panel. Make changes to the fields as necessary.

## Setting TWS Operation selection criteria

```

----- JOB/SCAN FOR TWS OPERATION SELECTION CRITERIA -----
COMMAND ==>

Review values and press ENTER to continue or END to return.

CURRENT PLAN ENDS  ==> 05/06/13 07.00      TWS Subsystem : TWS1
JOB NAME           ==> BR14JOB_           LU NAME      :
APPLICATION        ==> *                 Blank and wild cards OK
INPUT ARRIVAL      ==> __/__/__ __.____  Blank or YY/MM/DD HH.MM
OPERATION (WS NO.) ==> _____        Blank or WWWWW NNN
VARIABLE TABLE    ==> _____

SEARCH TWS CP      ==> YES                Yes or No
SEARCH TWS AD'S    ==> YES                Yes or No
PROVIDE NF ENTRY   ==> NO_               Yes or No (NF=Not Found)

```

6. Press Enter.



If you want to see a list of subsystems, type an asterisk (\*) in the *TWS Subsystem* field and press Enter.

7. JOB/SCAN displays the JOB/SCAN FOR TWS OPERATION LIST panel. It lists all the applications that TWS found in the Current Plan or Application Description data base that matched the supplied criteria (Application ID, Job-Name and so on). Select the appropriate entry with the **S** command.
8. Press Enter.

## The Application Selection panel

```

----- JOB/SCAN FOR TWS OPERATION LIST ----- Row 1 to 9 of 23
COMMAND ==>                                     SCROLL ==> PAGE

Use S to select. Press ENTER to process or END to return.

CURRENT PLAN ENDS  ==> 05/06/09 03.01      TWS SUBSYSTEM : TWS7
JOB NAME           ==> BR14JOB             LU NAME      :
APPLICATION        ==> *                 Blank for default
VARIABLE TABLE    ==> _____

Sel Application id  Operation Jobname   Input Arrival  Variable      Status
                  ws   no.           Date  Time     Table         Sorce

S  APPL1           DSS7 040 BR14JOB   05/03/28 02.00  AUSTIN#VARS   E   CP
_  APPL2           DSS7 040 BR14JOB   05/04/04 02.00  AUSTIN#VARS   W   CP
...

```

9. JOB/SCAN gives TWS the application ID and other criteria; TWS then processes the JCL and resolves the JCL variables. JOB/SCAN displays the SJL. See [Structured JCL Listing \(SJL\)](#) for details about the JOB/SCAN Structured JCL Listing (SJL).
10. When you have finished viewing the results, use the End command to return to step 7 where you may select another application or job or change the variable table.



- If you want to use a specific Variable Table, type in the name in the Variable Table field. If you want to see a list of available tables, type an asterisk (\*) in the field.
- For JOBS with promptable variables, JOB/SCAN displays the PROMPTABLE VARIABLE panel prior to validating the JOB. Be sure to use the End command to finish processing.
- If an application cannot be found, JOB/SCAN generates and displays a temporary application with the default application name and an NF (not found) code. After processing, the temporary application is deleted from the TWS application database.

#### The NOT FOUND temporary application ID

```

----- JOB/SCAN FOR TWS OPERATION LIST ----- Row 1 to 1 of 1
COMMAND ==>                                     SCROLL ==> PAGE

Use S to select. Press ENTER to process or END to return.

CURRENT PLAN ENDS ==> 05/06/09 03.01           TWS SUBSYSTEM : TWS7
JOB NAME           ==> BR14JOB                 LU NAME      :
APPLICATION        ==> *
VARIABLE TABLE   ==>                         Blank for default

Sel Application id  Operation Jobname   Input Arrival  Variable      Status
                  ws   no.           Date   Time    Table
_  APPL1           DSS7 001 BR14JOB   05/04/05 10.13

```

## Using the JTWS EDIT Macro from TWS Panels

Another way to employ JOB/SCAN is through the facilities TWS provides for editing or viewing a JCL member. There are several reasons for editing JCL while using TWS, for instance:

- From within the Current or Trial Plans you are confirming that this particular run of the job will resolve and scan successfully. This can be against either the saved version of the JCL or with a slightly modified version of the JCL being provided for this run only.
- From within the application database you are confirming that the JCL resolves correctly with the TWS application in which it is scheduled. This might be part of the development or maintenance life cycle of the job.

### Validating a Job in the Current Plan

*To use JTWS to start validation on a job any time you are in an EDIT or VIEW session*

1. Start in the TWS ISPF OPERATIONS PLANNING AND CONTROL panel. Select the MCP (Modify the Current Plan) option.

**TWS Operations Planning and Control panel**

```

----- OPERATIONS PLANNING AND CONTROL -----
Option ==> 5

Welcome to OPC. You are communicating with TWS7

Select one of the following options and press ENTER.

0 OPTIONS          - Define OPC dialog user parameters and options
1 DATABASE         - Display or update OPC data base information
2 LTP              - Long Term Plan query and update
3 DAILY PLANNING   - Produce daily plans, real and trial
4 WORK STATIONS    - Work station communication
5 MCP              - Modify the Current Plan
6 QCP              - Query the status of work in progress
7 OLD OPERATIONS   - Restart old operations from the DB2 repository

9 SERVICE FUNC     - Perform OPC service functions
10 OPTIONAL FUNC   - Optional functions
X EXIT             - Exit from the OPC dialog

```

2. TWS displays the MODIFYING THE CURRENT PLAN panel. Select the OPERATIONS option.

**Selecting to validate JCL in TWS operations**

```

----- MODIFYING THE CURRENT PLAN -----
Option ==> 3

Select one of the following:

1 ADD              - Add a new occurrence to the current plan
2 LIST             - List existing occurrences for further processing

3 OPERATIONS       - List existing operations for further processing
4 ERROR HANDLING   - Handle operations in error
5 WORK STATIONS    - Change status and open interval of work stations

6 JOB SETUP        - Prepare JCL for jobs in the current plan
7 SPECRES          - Special resource monitor
9 DEFINE EL        - Define alternative error list layouts

```

3. TWS displays the SELECTING OPERATIONS panel. This panel contains fields TWS uses as criteria for selecting operations. Type the name of the JOB in the JOBNAME field. Fill in or modify the other fields as appropriate.

### Selecting Operations panel

```

----- SELECTING OPERATIONS -----
Command ==>

Specify selection criteria below and press ENTER to create an operation list.

JOBNAME           ==> BR14JOB
FAST PATH         ==> N           Valid only along with jobname
                                Y Yes, N No
APPLICATION ID    ==> _____
OWNER ID         ==> _____
AUTHORITY GROUP   ==> _____
WORK STATION NAME ==> _____
PRIORITY         ==> -           Low priority limit
                                Y Yes, N No
MANUALLY HELD    ==> -           Status codes list:
STATUS           ==> _____ A R * S I C E W U and D

Input arrival in format YY/MM/DD HH.MM
FROM             ==> _____
TO               ==> _____
GROUP DEFINITION ==> _____
CLEAN UP TYPE    ==> _____ Types list: A M I N or blank
CLEAN UP RESULT  ==> _____ Results list: C E or blank
OP. EXTENDED NAME ==> _____

```

4. Press Enter.
5. TWS displays the Modifying Operations in the Current Plan panel.  
Select the application ID using the J line-command.

### Selecting an application and job from the Operations panel

```

----- MODIFYING OPERATIONS IN THE CURRENT PLAN -- Row 1 to 5 of 5
Command ==>                               Scroll ==> PAGE

Enter the GRAPH command above to view list graphically,
enter the HIST command to select operation history list, or
enter any of the following row commands:
J - Edit JCL                               M - Modify           B - Browse details
DEL - Delete Occurrence                    MH - Man. HOLD      MR - Man. RELEASE oper
O - Browse operator instructions            NP - NOP oper       UN - UN-NOP oper
EX - EXECUTE operation                     D - Delete Oper    RG - Remove from group
L - Browse joblog                          RC - Restart and CleanUp
FSR - Fast path SR                        FJR - Fast path JR
RI - Recovery Info

Row  Application id  Operat  Jobname  Input Arrival  Duration  Op  Depen  S Op
cmd  ws  no.  Date    Time  HH.MM.SS ST Su Pr  HN
'''  START050114    SYS1 030 BR14JOB 05/01/28 07.30 00.02.00 YN  0  0 E NN
J'''  XXXYUSR       SYS1 040 BR14JOB 05/01/24 02.00 00.01.00 NN  0  0 A NN
...

```



Single-character commands must be placed in the first space in the Row cmd field.

6. TWS displays the job in an Edit session. Type JTWS on the command line to validate the JCL.

## Invoking JOB/SCAN with JTWS

```

----- EDITING JCL FOR A COMPUTER OPERATION -----
Command ==> JTWS                               Scroll ==> PAGE

Edit JCL below and press END to finish or CANCEL to reject:

Application      : XXXYUSR                      INSERTED BY SYS1
Operation        : SYS1 40                     INSERTED BY SYS1
Status of operation : Arrived
Jobname          : BR14JOB                      JCL last updated by: SYSUSR

***** ***** Top of Data *****
010000 //BR14JOBA JOB
020000 //*%OPC SCAN
030000 //* OADID DEPENDENT VARIABLE (MODNAME)   = &MODNAME
040000 //* OCCURRENCE APPLICATION ID (OADID)    = &OADID
050000 //* OCCURRENCE OWNER (OADOWNER)         = &OADOWNER
060000 //* OCCURRENCE IA DATE TIME (OYMD1 OHMM) = &OYMD1 &OHMM
070000 //* OPERATION JOB NAME (OJOBNAME)       = &OJOBNAME
080000 //* OPERATION NUMBER (OOPNO)           = &OOPNO
090000 //* OPERATION LATEST START (OLYMD OLHMM) = &OLYMD &OLHMM
100000 //* OPERATION WORKSTATION (OWSID)       = &OWSID
110000 //* CURRENT DATE TIME (CYMMDD CHMM)    = &CYMMDD &CHMM
...

```

7. JOB/SCAN displays the JOB/SCAN FOR TWS OPERATION IDENTIFICATION panel. Modify the fields as necessary.
8. Press Enter to continue.

## Setting validation criteria for JOB/SCAN for TWS

```

----- JOB/SCAN FOR TWS OPERATION IDENTIFICATION -----
COMMAND ==>

Review values and press ENTER to continue or END to return.

CURRENT PLAN ENDS ==> 05/02/03 17.00

JOB NAME           ==> BR14JOB
APPLICATION        ==> XXXYUSR
INPUT ARRIVAL      ==> 05/01/24 02.00
OPERATION (WS NO.) ==> SYS1 040
VARIABLE TABLE    ==> USR321#VARS

```



- For JOBS with promptable variables, JOB/SCAN displays the PROMPTABLE VARIABLE panel prior to validating the JOB.
- If you want to use a specific variable table, type in the name in the Variable Table field. If you want to see a list of available tables, type an \* in the field.
- JOB/SCAN processes the JCL and displays the Structured JCL Listing. Press End to return to Modifying Operations in the Current Plan panel.

## Validating a Job with JTWS from an Application

### To do a validation on application JCL

1. From the TWS OPERATIONS PLANNING AND CONTROL menu, use the 1;4;1 fast-path to browse or the 1;4;3 fast-path to edit the JCL respectively.

### Selecting TWS Application update

```

----- OPERATIONS PLANNING AND CONTROL -----
Option ==> 1;4;3

Welcome to OPC. You are communicating with TWS7

Select one of the following options and press ENTER.

0 OPTIONS          - Define OPC dialog user parameters and options
1 DATABASE         - Display or update OPC data base information
2 LTP              - Long Term Plan query and update
3 DAILY PLANNING  - Produce daily plans, real and trial
4 WORK STATIONS   - Work station communication
5 MCP              - Modify the Current Plan
6 QCP              - Query the status of work in progress
7 OLD OPERATIONS  - Restart old operations from the DB2 repository
...

```

2. JOB/SCAN displays the SPECIFYING APPLICATION LIST CRITERIA panel.

### Specifying Application List Criteria panel

```

----- SPECIFYING APPLICATION LIST CRITERIA -----
Command ==>

Specify selection criteria below and press ENTER to create an application list.

APPLICATION ID    ==> TPIAPP2_____
TYPE              ==> A           A = Application, G = Group AD (blank=both)
JOBNAME           ==> OPCTST3_
OWNER ID          ==>
STATUS            ==> -           A = active, P = pending (blank=both)
LOW PRIORITY      ==> -           Low priority limit (1-9)
AUTHORITY GROUP ID ==>
Valid any time in period below:
  FROM DATE       ==> _____ Date in the format YY/MM/DD
                  (blank = no starting limit of period)
  TO DATE         ==> _____ Date in the format YY/MM/DD
                  (blank = no ending limit of period)
CALENDAR ID       ==> _____
WORK STATION      ==> _____
GROUP DEFINITION  ==> _____
OP. EXTENDED NAME ==> _____
OP. SE NAME       ==> _____

```

3. Type values for the APPLICATION ID, APPLICATION TYPE, JOBNAME and other criteria. Press Enter.
4. On the LIST OF APPLICATIONS panel, use the M command to select the application to validate.

### Selecting an application from the TWS LIST OF APPLICATIONS panel

```

----- LIST OF APPLICATIONS ----- Row 1 to 1 of 1
Command ==>                               Scroll ==> PAGE

Enter the CREATE command above to create a new application, or,
enter the GRAPH command above to view the list graphically, or,
enter any of the row commands below:
B - Browse, M - Modify, C - Copy, D - Delete,
P - Print, A - Calculate and print run days,
L - Modify LTP (external dependencies are not resolved)

Row Application          Valid      T S Owner id
cmd id                  text      from date
M  TPIAPP2              04/09/09  A A SYSUSR
***** Bottom of data *****

```

- At the TWS panel for browsing or modifying an application, update the values in the fields and use the OPER command to access the operations of the application.

### Using the OPER command to list operations for an application

```

----- MODIFYING AN APPLICATION -----
Command ==> OPER

Enter/Change data below:
Enter the RUN command above to select run cycles or enter the OPER command
to select operations.

Application id          : TPIAPP2
Valid from - to       : 05/06/03 - 71/12/31

APPLICATION TEXT ==> _____ Descriptive text
TYPE             ==> A           A = Application, G = Group definition
OWNER ID         ==> SYSUSR_____
OWNER TEXT       ==> _____
PRIORITY         ==> 3           A digit 1 to 9 , 1=low, 8=high, 9=urgent
VALID FROM       ==> 05/06/03   Date in the format YY/MM/DD
STATUS           ==> A           A - Active, P - Pending
AUTHORITY GROUP ID ==> _____ Authorization group ID
CALENDAR ID      ==> _____ For calculation of work and free days
GROUP DEFINITION ==> _____ Group definition id

Last updated by SYSUSR on 05/03/30 at 13.31

```

- TWS displays the Operations panel. Use the J command to invoke the JOB/SCAN Edit tool on the job.
- Press Enter.



**Selecting an operation for editing JCL from the TWS OPERATIONS panel**

```

----- OPERATIONS ----- Row 1 to 1 of 1
Command ==> Scroll ==> PAGE

Enter/Change data in the rows, and/or enter any of the following
row commands:
I(nn) - Insert, R(nn),RR(nn) - Repeat, D(nn),DD - Delete
S - Select operation details, J - Edit JCL
Enter the PRED command above to include predecessors in this list, or,
enter the GRAPH command to view the list graphically.

Application           : TPIAPP2

Row  Oper      Duration  Job name  Operation text
cmd  ws   no.   HH.MM.SS
J    DSS7 010   00.00.10  BR14JOB_

```

8. TWS passes control to the JOB/SCAN FOR TWS EDIT JCL panel. On this panel, use command 1 to browse or edit the JCL (JOB/SCAN displays the appropriate panel title for the 1;4;1 or 1;4;3 fast-path).
9. Press Enter.

**Invoking Browse or Edit on the job**

```

----- JOB/SCAN FOR TWS - EDIT JCL -----
Option ==> 1

Subsystem Name: TWS7
Application   : TPIAPP2
Workstation  : DSS7
Operation No : 010
Job Name     : BR14JOB_

0 OPTIONS    - Specify Alternate JCL Edit TOOL

1 EDIT      - EDIT JCL from EQQJBLIB
X EXIT

NOTE       : From EDIT use command JTWS to validate BR14JOB
            with resolved variables using JOB/SCAN from
            ASG, Inc.

```

10. JOB/SCAN displays the member in a View or Edit session. Type the JTWS Edit macro name on the command line.
11. Press Enter.

**Invoking JOB/SCAN on the job**

```

VIEW          TWS.V8R2M0.JOBLIB(BR14JOB) - 01.01          Columns 00001 00072
Command ==> JTWS          Scroll ==> PAGE
***** ***** Top of Data *****
000100 //BR14JOBA JOB
000200 // *%OPC SCAN
000300 // * OADID DEPENDENT VARIABLE (MODNAME)           = &MODNAME
000400 // * OCCURRENCE APPLICATION ID (OADID)            = &OADID
000500 // * OCCURRENCE OWNER (OADOWNER)                 = &OADOWNER
000600 // * OCCURRENCE IA DATE TIME (OYMD1 OHHMM)       = &OYMD1 &OHHMM
000700 // * OPERATION JOB NAME (OJOBNAME)              = &OJOBNAME
000800 // * OPERATION NUMBER (OOPNO)                   = &OOPNO
001900 // * OPERATION LATEST START (OLYMD OLHHMM)      = &OLYMD &OLHHMM
001000 // * OPERATION WORKSTATION (OWSID)              = &OWSID
001100 // * CURRENT DATE TIME (CYMMDD CHHMM)           = &CYMMDD &CHHMM
...

```

12. JOB/SCAN displays the JOB/SCAN FOR TWS OPERATION IDENTIFICATION panel. Confirm and adjust as necessary, *Input Arrival* and *Variable Table* values.
13. Press Enter.

**Setting operation identification criteria for the JCL**

```

----- JOB/SCAN FOR TWS OPERATION IDENTIFICATION -----
COMMAND ==>

Review values and press ENTER to continue or END to return.

CURRENT PLAN ENDS ==> 05/06/25 03.01

JOB NAME           ==> BR14JOB
APPLICATION        ==> TPIAPP2
INPUT ARRIVAL      ==> 05/06/03 09.16
OPERATION (WS NO.) ==> DSS7 010
VARIABLE TABLE    ==> AUSTIN#VARS

```

14. JOB/SCAN displays the Structured JCL Listing (SJL).

When you have finished viewing the SJL, use the End command to return to the Edit session. If you make additional changes to the JCL in the Edit session, then repeat steps 5 and 6 to validate your changes.

## Validate Members from the Validation Panel



ASG recommends validating only one member at a time through the JOB/SCAN JCL Validation panel while using the TWS interface; this assures that each job is validated with the appropriate TWS application and variable table.

**To validate JCL containing TWS variables**

1. Start JOB/SCAN. When the system displays the ASG-JOB/SCAN Primary Menu.

**Selecting JCL validation**

```

----- ASG-JOB/SCAN PRIMARY MENU -----
SELECTION ==> 1

    0 - Hardcopy Output and JOB Statement Definition
    1 - Validate JCL
    2 - Change / Reformat JCL
    3 - Processing Options
    4 - Selection List Maintenance
    ...

USERID - SYSUSR
PREFIX - SYSUSR
DATE   - 2006/02/13
TIME   - 10:32
APPLID - JS

```

2. Select *Validate JCL*.
3. Press Enter.
4. JOB/SCAN displays the JOB/SCAN JCL Validation panel.

**Selecting foreground validation mode**

```

----- JOB/SCAN - JCL VALIDATION -----
COMMAND ==> F

    F - Foreground execution mode      P - Processing Options
    B - Batch execution mode           E - Edit JCL member
                                      BR - Browse JCL member

Input Library ==> SYSUSR.JCL
Input Type    ==> J (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ==> PDS (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> MYJOB
Exclude Members ==>
First Proclib ==>
Proclib Type  ==> (PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
Report Heading ==>
Display        ==> Y (Yes/No/ErrorsOnly) Browse/View ==> B (B/V)
Hardcopy       ==> N (Yes/No/ErrorsOnly/SpaceOnly/JobErrors)
Options in Effect Listing ==> N (Y/N) SJL PDSOUT ==> N (Y/N)

Selection Exit ==> Parm ==>

```

5. Use the F command to begin processing.
6. JOB/SCAN displays the JOB/SCAN Advisory panel.

**Continuing through the Advisory panel**

```

----- JOB/SCAN FOR TWS ADVISORY -----
COMMAND ==> Y

    Y - to continue
    N - to return

JOB/SCAN updates the occurrence JCL in the JS FILE. After
TWS resolves the variables, JOB/SCAN restores this Job
to its original contents.

_ Enter "X" to suppress this panel in the future.

```

7. Type Y on the command line to continue processing.
8. JOB/SCAN displays the JOB/SCAN for TWS Operation Selection Criteria panel. This example shows a wildcard in the application field; this causes JOB/SCAN to display the JOB/SCAN for TWS Operation List.

### Supplying Operation Selection Criteria

```

----- JOB/SCAN FOR TWS OPERATION SELECTION CRITERIA -----
COMMAND ==>

Review values and press ENTER to continue or END to return.

CURRENT PLAN ENDS ==> 06/02/20 12.00          TWS Subsystem : TWS6
JOB NAME           ==> BR14JOB                LU NAME      :
APPLICATION        ==> *                    Blank and wild cards OK
INPUT ARRIVAL      ==> __/__/__  __. __      Blank or YY/MM/DD HH.MM
OPERATION (WS NO.) ==> _____           Blank or WWWWW NNN
VARIABLE TABLE   ==> _____

SEARCH TWS CP      ==> YES                   Yes or No
SEARCH TWS AD'S    ==> NO                   Yes or No
PROVIDE NF ENTRY   ==> NO                   Yes or No (NF=Not Found)

```

9. JOB/SCAN displays the JOB/SCAN for TWS Operation List panel.

### Selecting an Operation

```

----- JOB/SCAN FOR TWS OPERATION LIST ----- Row 1 to 9 of 31
COMMAND ==>                                SCROLL ==> PAGE

Use S to select. Press ENTER to process or END to return.

CURRENT PLAN ENDS ==> 06/02/20 12.00          TWS SUBSYSTEM : TWS6
JOB NAME           ==> BR14JOB                LU NAME      :
APPLICATION        ==> *                    Blank for default
VARIABLE TABLE   ==> _____

Sel Application id  Operation Jobname   Input Arrival  Variable      Status
                   ws   no.           Date   Time     Table
                   _____
-   JWA1             CPU6 010 BR14JOB   05/12/19 15.16      E   CP
-   JWA1             CPU6 010 BR14JOB   05/12/27 12.00      E   CP
S   SYSUSR          CPU1 001 BR14JOB   05/09/06 11.00      E   CP
-   SYSUSR          CPU1 001 BR14JOB   05/09/13 11.00      A   CP
-   SYSUSR          CPU1 001 BR14JOB   05/09/20 11.00      A   CP

```

10. Select an application/operation.
11. Press Enter. JOB/SCAN displays the SJL with the TWS variables resolved.

## Validate a Single Member in Batch

### To process a single member in batch

1. Make a copy of the JSMVS member (from the SJSSAMP library). See figure below.
2. Change the JOB name and add appropriate fields and parameters to JOB card.
3. Add the input data set.

4. Add the member name.
5. Add these DDs after the REPORT OPTIONS statement:

```
//STEP01.JLSCHO DD SYSOUT=*
//STEP01.JLSCH DD *
```

6. Add these DDs if they have not been added to JOB/SCAN JS6PROC:

```
//EQQEMSG DD SYSOUT=*
//EQQMLOG DD SYSOUT=*
//EQQDUMP DD SYSOUT=*
//EQQMLIB DD DISP=SHR,DSN=TWS.XXXXXXX.SEQMSG0 <== Modify*
```

7. Add this control card directly after the JLSCH DD:

```
TWS ADID(applicationID) -
    OPRNUM(operationnumber) -
    IADATE(arrivaldate) IATIME(arrivaltime) -
    JCLVTAB(tablename)
```

This table lists the control card values (if any value contains delimiters or spaces place quotes around it):

Control Card	Description
ADID(applicationID)	Application description ID name.
WSN(workstationname)	(Optional) Workstation name.
OPRNUM(operationnumber)	Operation number; a positive integer 1-255.
IADATE(arrivaldate)	Input arrival date (YYMMDD).
IATIME(arrivaltime)	Input arrival time (HHMM).
JCLVTAB(tablename)	Optional) TWS variable table to use.

Precede the first card with TWS. Place a continuation character (-) at the ends of lines to continue to the next line.

## Modifying batch JCL to validate a single member

```
//MYJOB JOB (ACCT), 'PROGRAMMER NAME'
//*****
//* EXECUTE JOB/SCAN (MVS SYSTEMS)
//* REF: SAMPLIB (JSMVS)
//*
//* NOTE: MODIFY JOB STATEMENT AND ITEMS INDICATED BY ARROWS <==
//*****
//JSSTEP1 EXEC JS6PROC,PRM=S
//SYSIN DD *
INPUT SELECTION
    DSN=YOUR.JCLLIB                <== INPUT JCL LIBRARY
    SELECT=YOURJOB                 <== MEMBER TO SCAN
    SOURCE TYPE=PDS                <== TYPE OF INPUT LIB
REPORT OPTIONS
    OPTIONS IN EFFECT =PRINT       <== PRINT OPE REPORT?
    STRUCTURED JCL LISTING=PRINT   <== PRINT SJL REPORT?
//STEP01.JLSCHO DD SYSOUT=*
//STEP01.JLSCH DD *
    TWS ADID(MYAPPL) OPRNUM(003) IADATE(100325) IATIME(1200)
```

## Validating a Plan

This topic describes how to validate the JCL for a TWS plan.

- [Validate JCL for the Current Plan](#)
- [Validate JCL for the Trial Plan](#)

## Validate JCL for the Current Plan

There are two approaches to validating the JCL for the Current Plan:

- from JOB/SCAN.
- from TWS.



JOB/SCAN relies on the TWS planned start date/time to order jobs for validation. TWS special resources, when used for planning, can cause incorrect planned start date/time to be generated. This might cause jobs to be validated in the wrong order resulting in incorrect validation errors. (IBM solution #37574 describes this problem.)

### Validate Current Plan JCL from JOB/SCAN

#### *To begin the validation for the Current Plan from JOB/SCAN*

1. Select the Validate TWS for z/OS Scheduler JCL option from the ASG-JOB/SCAN Primary Menu panel to begin.
2. Press Enter.

**Validate Current Plan with ASG-JOB/SCAN Primary Menu**

```

----- ASG-JOB/SCAN PRIMARY MENU -----
SELECTION ==> S

    0 - Hardcopy Output and JOB Statement Definition
    1 - Validate JCL
    2 - Change / Reformat JCL
    3 - Processing Options
    4 - Selection List Maintenance
    5 - Assign Alternate Options Members
    6 - Product Administration Menu
    7 - JOB/SCAN Usage Report
    S - Validate TWS for z/OS Scheduler JCL
    X - Exit

USERID - SYSUSR
PREFIX - SYSUSR
DATE   - 04/11/2005
TIME   - 09:02
APPLID - JS

```



Because a different name might have been chosen during the JOB/SCAN installation process, the name of the option on the ASG-JOB/SCAN Primary Menu may differ from the one shown here.

3. Continue with [Entering Data for Validating JCL for the Current Plan](#).

**Validate Current Plan JCL from TWS****To begin the validation for the Current Plan from TWS**

1. Use the **JVAL** command to validate the JCL for the Current Plan (see [JVAL Command](#)) from any TWS panel. TWS invokes the JOB/SCAN for TWS - JCL Validation panel.
2. Press Enter.

**Validating the Current Plan**

```

----- OPERATIONS PLANNING AND CONTROL -----
Option ==> JVAL

Welcome to OPC. You are communicating with TWS7

Select one of the following options and press ENTER.

0 OPTIONS           - Define OPC dialog user parameters and options
1 DATABASE          - Display or update OPC data base information
2 LTP               - Long Term Plan query and update
3 DAILY PLANNING    - Produce daily plans, real and trial
4 WORK STATIONS     - Work station communication
5 MCP              - Modify the Current Plan
6 QCP              - Query the status of work in progress
7 OLD OPERATIONS    - Restart old operations from the DB2 repository

9 SERVICE FUNC      - Perform OPC service functions
10 OPTIONAL FUNC    - Optional functions
X EXIT              - Exit from the OPC dialog

```

3. Continue with [Entering Data for Validating JCL for the Current Plan](#).

## Entering Data for Validating JCL for the Current Plan

### To finish validating JCL for the current plan

1. JOB/SCAN displays the JOB/SCAN FOR TWS - JCL Validation panel. Select CURRENT PLAN.
2. Press Enter.

### JOB/SCAN for TWS Validation main menu

```

----- JOB/SCAN FOR TWS - JCL VALIDATION -----
Option ==> 2

Welcome to JOB/SCAN for TWS. You are communicating with TWS6

To validate JCL in execution sequence with TWS variable resolution
select one of the following options and press ENTER.

1 APPLICATION      - JCL Validation for an application
2 CURRENT PLAN    - JCL Validation from CP
3 TRIAL PLAN      - JCL Validation from a TRIAL PLAN
X  EXIT

```

JOB/SCAN displays the JOB/SCAN for TWS Current Plan Validation panel.

3. Type values into the fields, as necessary (these fields are “sticky”, meaning that they retain previously-filled-in values).
4. Choose a command:
  - Use command **s** to submit the job for validation.

**Or**

  - Use command **⌘** to edit the JCL. Then use the SUB command to submit the job for validation.



For more information about the TWS parameters JOB/SCAN puts into the JCL see [JCL Parameters for the JVAL Command JCL](#) .

5. JOB/SCAN writes the bold values to the JCL before submitting it or displaying it for Edit.
6. After submitting, monitor the output hold queue for status and output for the job.



**JOB/SCAN for TWS Current Plan Validation panel**

```

----- JOB/SCAN FOR TWS - CURRENT PLAN VALIDATION -----
Command ==>

          S to submit the validation JOB   E to edit the validation JOB

Specify selection criteria below:
TWS SUBSYSTEM NAME ==> TWS7

JOBNAME           ==> BR* _____
APPLICATION ID    ==> X* _____
OWNER ID          ==> XXX* _____
AUTHORITY GROUP ID ==> * _____
WORK STATION NAME ==> * _____
PRIORITY          ==> 1 _____      Low priority limit
STATUS           ==> A _____      Status Codes List:
                                           A R * S I C E W U and D

Input Arrival in format YY/MM/DD HH.MM
FROM              ==> 05/02/07 23.59
TO                ==> 05/02/08 11.59
GROUP DEFINITION ==> MYGROUP _____

SELECTION EXIT   ==> _____ Parm ==> _____
STANDARDS NAME  ==> _____ Parm ==> _____

```

## Validate JCL for the Trial Plan

There are two approaches to validating the JCL for the Trial Plan:

- from JOB/SCAN.
- from TWS.

### Validate Trial Plan from JOB/SCAN

#### *To begin the validation for the Trial Plan*

1. Select the Validate TWS for z/OS Scheduler JCL option from the ASG-JOB/SCAN Primary Menu panel to begin.
2. Press Enter.

#### Validating Trial Plan with ASG-JOB/SCAN Primary Menu

```

----- ASG-JOB/SCAN PRIMARY MENU -----
SELECTION ==> S

0 - Hardcopy Output and JOB Statement Definition
1 - Validate JCL
2 - Change / Reformat JCL
3 - Processing Options
4 - Selection List Maintenance
5 - Assign Alternate Options Members
6 - Product Administration Menu
7 - JOB/SCAN Usage Report
S - Validate TWS for z/OS Scheduler JCL
X - Exit

USERID - SYSUSR
PREFIX - SYSUSR
DATE - 04/11/2005
TIME - 09:02
APPLID - JS

```

JOB/SCAN displays the JOB/SCAN for TWS - JCL Validation panel.

3. Continue on [Entering Data for Validating JCL for the Trial Plan](#).

### Validate Trial Plan from TWS

#### To begin the validation for the Trial Plan from TWS

1. Use the JVAL command to validate the JCL for the Trial Plan from any TWS panel. (See [JVAL Command](#) for more about the JVAL command.)
2. Press Enter.

#### Validating trial plan

```

----- OPERATIONS PLANNING AND CONTROL -----
Option ==> JVAL

Welcome to OPC. You are communicating with TWS7

Select one of the following options and press ENTER.

0 OPTIONS          - Define OPC dialog user parameters and options
1 DATABASE         - Display or update OPC data base information
2 LTP              - Long Term Plan query and update
3 DAILY PLANNING   - Produce daily plans, real and trial
4 WORK STATIONS    - Work station communication
5 MCP              - Modify the Current Plan
6 QCP              - Query the status of work in progress
7 OLD OPERATIONS   - Restart old operations from the DB2 repository

9 SERVICE FUNC     - Perform OPC service functions
10 OPTIONAL FUNC   - Optional functions
X EXIT            - Exit from the OPC dialog

```

3. Continue on [Entering Data for Validating JCL for the Trial Plan](#).

### Entering Data for Validating JCL for the Trial Plan

#### To enter data and finish validating JCL for the Trial Plan

1. Select TRIAL PLAN.
2. Press Enter.

#### JOB/SCAN for TWS Validation main menu

```

----- JOB/SCAN FOR TWS - JCL VALIDATION -----
Option ==> 3

Welcome to JOB/SCAN for TWS. You are communicating with TWS6

To validate JCL in execution sequence with TWS variable resolution
select one of the following options and press ENTER.

1 APPLICATION      - JCL Validation for an application
2 CURRENT PLAN     - JCL Validation from CP
3 TRIAL PLAN       - JCL Validation from a TRIAL PLAN
X EXIT            -

```

JOB/SCAN displays the JOB/SCAN for TWS TRIAL PLAN VALIDATION panel.

3. Type the values into the fields, as necessary (these fields might contain previously-filled-in values).
4. Choose a command:
  - Use command **S** to submit the job for validation.

**Or**

  - Use command **E** to edit the JCL. Then use the SUB command to submit the job for validation.



For more information about the TWS parameters JOB/SCAN puts into the JCL see [JCL Parameters for the JVAL Command JCL](#) .

5. JOB/SCAN writes the bold values to the JCL before submitting it or displaying it for Edit.
6. After submitting, monitor the output hold queue for status and output for the job.

#### Setting criteria for Trial plan validation

```

----- JOB/SCAN FOR TWS - TRIAL PLAN VALIDATION -----
Command ==>

      S to submit the validation JOB   E to edit the validation JOB

Specify selection criteria below:

TWS SUBSYSTEM NAME  ==> TWS7
WORK FILE PREFIX    ==> MY.WORK.FILE_____

APPLICATION ID      ==> X*_____ - _____ (range)
OWNER ID            ==> XXX*_____
AUTHORITY GROUP ID ==> *_____
PRIORITY            ==> 1_____

Input Arrival in format  YY/MM/DD  HH.MM
FROM                     ==> 05/02/07  23.59
TO                       ==> 05/02/08  11.59
GROUP DEFINITION        ==> MYGROUP_____

SELECTION EXIT  ==> _____ Parm ==> _____
STANDARDS NAME  ==> _____ Parm ==> _____

```

## Validate Application JCL

This topic describes how to validate the JCL for a TWS plan.

- [Validate Application JCL from TWS](#)
- [Application JCL Validation from JOB/SCAN](#)

## Validate Application JCL from TWS

Use the JVAL Command from any TWS ISPF panel. There are two TWS contexts for using the JVAL command to validate JCL for applications: Generic and Specific.

### Generic Context

From anywhere within TWS, you initiate JCL validation with JOB/SCAN by using the JVAL command (see [JVAL Command](#)

for details about JVAL). In the Generic context, you supply the ID of the application.

### To process JCL for an application while outside of the application context

1. Type the JVAL command on the commnd line of a TWS panel.

#### Entering the JVAL command on a TWS panel

```
----- DEFINING OPC PARAMETERS AND OPTIONS -----
Option ==> JVAL

Select one of the following:

0 REINIT          - Re-initialize the application profile values
1 SUBSYSTEM      - Set or change name of Subsystem and Server LU
2 DATE           - Specify date/time formats and default calendar
3 COLOR          - Specify panel color and highlight attributes
4 ISPF OPTIONS   - Specify ISPF/PDF options
5 AD/OI CHECKS   - Specify AD/OI consistency checks
6 JCL EDIT       - Specify JCL edit tool
7 CLEANUP CHECK  - Specify check option for Automatic Cleanup type
```

2. Press Enter. TWS invokes JOB/SCAN which displays the JOB/SCAN FOR TWS - JCL VALIDATION panel.
3. Choose APPLICATION. For example:

#### The JOB/SCAN FOR TWS JCL VALIDATION top menu

```
----- JOB/SCAN FOR TWS - JCL VALIDATION -----
Option ==> 1

Welcome to JOB/SCAN for TWS. You are communicating with TWS6

To validate JCL in execution sequence with TWS variable resolution
select one of the following options and press ENTER.

1 APPLICATION      - JCL Validation for an application
2 CURRENT PLAN    - JCL Validation from CP
3 TRIAL PLAN      - JCL Validation from a TRIAL PLAN
X EXIT
```

4. Press Enter. JOB/SCAN invokes the JOB/SCAN FOR TWS - VALIDATE APPLICATION panel.

#### Entering an Application ID

```
----- JOB/SCAN FOR TWS - VALIDATE APPLICATION -----
Command ==>

      S to submit the validation JOB   E to edit the validation JOB

Specify selection criteria below:

TWS SUBSYSTEM NAME  ==> TWS6
SIMULATED IA TIME   ==> 06/02/15  11.59   In format YY/MM/DD HH.MM

APPLICATION ID      ==> MYUSERS

SELECTION EXIT  ==> _____ Parm ==> _____
STANDARDS NAME  ==> _____ Parm ==> _____
```

5. Furnish an application name and any standards exit and selection exit names desired.
6. Choose a command:
  - Use command `s` to submit the job for validation.

**Or**

  - Use command `E` to edit the JCL. Then use the SUB command to submit the job for validation.



For more information about the TWS parameters JOB/SCAN puts into the JCL see [JCL Parameters for the JVAL Command JCL](#) .

7. After submitting, monitor the output hold queue for status and output for the job.

### Specific Context

When working with TWS application, the JVAL command can validate JCL for that application. JOB/SCAN obtains the application ID from the TWS panel.

#### *To specify a particular application*

1. From either the BROWSING AN APPLICATION or MODIFYING AN APPLICATION panels in TWS, use the JVAL command (without parameters). TWS invokes JOB/SCAN to display the JOB/SCAN FOR TWS VALIDATE APPLICATION panel, inserting details from the application.

#### Entering the JVAL command in an Application context

```

----- BROWSING AN APPLICATION -----
Command ==> JVAL

Enter the RUN command above to select run cycles or enter the OPER command
to select operations.

Application           : MYUSERS           USER text
Type                  : Application
Owner                 : USER SMITHSON
Priority               : 1
Valid from - to      : 04/10/19 - 15/12/31
Authority group id   :
Status                : Active
Calendar id          :
Group definition      :

Total number of:
Run cycles            : 0
Operations            : 1
External predecessors : 0

```

JOB/SCAN displays the VALIDATE APPLICATION panel for the application.

**Pre-filled data fields**

```

----- JOB/SCAN FOR TWS - VALIDATE APPLICATION -----
Command ==> S

      S to submit the validation JOB   E to edit the validation JOB

Specify selection criteria below:

TWS SUBSYSTEM NAME  ==> TWS7
SIMULATED IA TIME   ==> 05/04/11   09.42   In format YY/MM/DD HH.MM

APPLICATION ID       ==> MYUSERS_____

SELECTION EXIT      ==> _____ Parm ==> _____
STANDARDS NAME      ==> _____ Parm ==> _____

```

2. Enter desired standards exit and selection exit names.
3. Choose a command:
  - Use command **s** to submit the job for validation.

**Or**

  - Use command **E** to edit the JCL. Then use the SUB command to submit the job for validation.



For more information about the TWS parameters JOB/SCAN puts into the JCL see [JCL Parameters for the JVAL Command JCL](#) .

4. After submitting, monitor the output hold queue for status and output for the job.

## Application JCL Validation from JOB/SCAN

### To validate application JCL from JOB/SCAN

1. Select the Validate TWS for z/OS Scheduler JCL option from the ASG-JOB/SCAN Primary Menu panel to begin.

### Validate Application JCL with ASG-JOB/SCAN Primary Menu

```

----- ASG-JOB/SCAN PRIMARY MENU -----
SELECTION ==> S

      0 - Hardcopy Output and JOB Statement Definition
      1 - Validate JCL
      2 - Change / Reformat JCL
      3 - Processing Options
      4 - Selection List Maintenance
      ...
      S - Validate TWS for z/OS Scheduler JCL
      X - Exit

      USERID - SYSUSR
      PREFIX - SYSUSR
      DATE - 04/11/2005
      TIME - 09:02
      APPLID - JS

```

JOB/SCAN displays the JOB/SCAN FOR TWS - JCL VALIDATION panel.

2. Select APPLICATION.
3. Press Enter. JOB/SCAN displays the VALIDATE APPLICATION panel.

#### JOB/SCAN FOR TWS VALIDATION main menu

```

----- JOB/SCAN FOR TWS - JCL VALIDATION -----
Option ==> 1

Welcome to JOB/SCAN for TWS. You are communicating with TWS6

To validate JCL in execution sequence with TWS variable resolution
select one of the following options and press ENTER.

1  APPLICATION      - JCL Validation for an application
2  CURRENT PLAN    - JCL Validation from CP
3  TRIAL PLAN      - JCL Validation from a TRIAL PLAN
X  EXIT

```

4. In the JOB/SCAN FOR TWS - VALIDATE APPLICATION panel ([JOB/SCAN FOR TWS VALIDATE APPLICATION panel](#)), type the values for the fields, as necessary (these fields are “sticky”, meaning that they retain previously-filled-in values).
5. Choose a command:
  - Use command **s** to submit the job for validation.

**Or**

  - Use command **E** to edit the JCL. Then use the **SUB** command to submit the job for validation.



For more information about the TWS parameters JOB/SCAN puts into the JCL see [JCL Parameters for the JVAL Command JCL](#) .

#### JOB/SCAN FOR TWS VALIDATE APPLICATION panel

```

----- JOB/SCAN FOR TWS - VALIDATE APPLICATION -----
Command ==>

          S to submit the validation JOB   E to edit the validation JOB

Specify selection criteria below:

TWS SUBSYSTEM NAME  ==> TWS7
SIMULATED IA TIME   ==> 05/03/15   14.30   In format YY/MM/DD HH.MM

APPLICATION ID      ==> _____

SELECTION EXIT      ==> _____ Parm ==> _____
STANDARDS NAME      ==> _____ Parm ==> _____

```

6. After submitting, monitor the output hold queue for status and output for the job.

## Installation

### Version Support for New JOB/SCAN Customers

Starting with version 6.2.8A, enhanced support for the TWS for z/OS Scheduler is fully integrated into JOB/SCAN. It is not necessary to install additional support (from the JOB/SCAN TWS for z/OS FunctionPack) to enhance the TWS for z/OS scheduler interface. Follow the instructions in this topic to install the interface.

### Version Support for Existing JOB/SCAN Customers

Starting with version 6.2.8A, enhanced support for the TWS for z/OS Scheduler is fully integrated into JOB/SCAN. It is no longer necessary to concatenate the JOB/SCAN TWS for z/OS FunctionPack data sets ahead of JOB/SCAN product data sets. If you previously installed the FunctionPack, first remove these concatenations prior to installing the TWS for z/OS Scheduler Interface and rename or archive the FunctionPack data sets. Then, continue with the instructions in this topic to install the interface.

- [Installation Checklist](#)
- [Security](#)
- [Installation Procedure](#)
- [Providing Access to JOB/SCAN for TWS](#)
- [Activate TWS Scheduler Interface](#)
- [Reporting on TWS Special Resources \(Optional\)](#)
- [Verify Installation](#)

## Installation Checklist

Prior to beginning the installation, it is vitally important for the installer to obtain specific information about three central components of the interface support. These components, and some of the potential information required, are:

- JOB/SCAN product library data set names.
- APF-authorized load library name.
- JOB/SCAN data set naming convention.
- Tivoli Workload Scheduler for z/OS (TWS) - configuration information including subsystem names, server LU names, EQQJBLIB DD data set names, TWS for z/OS software product library high-level qualifiers, TWS for z/OS database data set high level qualifiers.

## Security

The JOB/SCAN TWS for z/OS interface will require read access for most TWS for z/OS fixed resources including AD, CL, CP, JS, JV, LT, PR, RD, & WS.

The JTWS Edit macro processing must insert or temporarily replace JCL in the JS file for proper variable resolution by TWS for z/OS to occur. Therefore, all JTWS Edit macro users require Update access to the JS resource.



To resolve variables in a job, that job must be associated with an application. When that application is Not Found (NF) in the application description data base (AD), the JTWS Edit macro must temporarily add the application to the AD. The Owner ID for this application is "SYS1". The operation extended name is "INSERTED BY SYS1". The authority group and group definition ID are blanks. This new application will be deleted once variable resolution is complete. JTWS Edit macro users who utilize the Not Found (NF) functionality will require Update access to the AD resource.

The JOB/SCAN for TWS interface JCL edit plug in panel (J00PTS00) will allow editing of JCL directly from the data sets specified in the EQQJBLIB ddname data sets (the controller Job library) as specified in the REXX CLIST member J00CTWS. Users who use this JCL edit plug-in require Update access to the specified data sets if they intend to modify the JCL; otherwise, Read access is sufficient.



This JCL edit plug in does not obtain or modify JCL in the JS file. If records are deleted or modified with this JCL edit plug in, the changes are saved in the specified dataset.

For additional information, see *IBM Tivoli Workload Scheduler for z/OS Customization and Tuning (SC32-1265)*.

## Installation Procedure



The installer must be familiar with making REXX Exec modifications.

Familiarize yourself with the installation by reading through this topic, noting the information requested in the spaces provided. Then, install the interface including the verification in [Verify Installation](#).

When you complete the installation, keep your annotated topic for future reference.

### Make JOB/SCAN LOAD library APF-Authorized for Batch Processing

In order to properly communicate with TWS for z/OS in the batch environment, the JOB/SCAN LOAD library must be an APF-authorized library. See - [Running the Product in an APF-authorized Environment](#).

### Make JS6PROC Available to Users

The procedure JS6PROC runs JOB/SCAN in batch. The JOB/SCAN skeleton library (SJJSLIB) members J00STS2A, J00STS2B and J00STS4A invoke it. If this proc is in a PROCLIB accessible to all users, no changes to these members are necessary. Skip to [Providing Access to JOB/SCAN for TWS](#).

If JS6PROC is not accessible to all users, add (uncomment) a JCLLIB statement to the skeleton members (J00STS2A, J00STS2B, J00STS4A) substituting the data set name of the JS6PROC PROCLIB for &XHLQ. . &NODE.

```
&XHLQ. . &NODE. _____
```

### Activating JCLLIB for JOB/SCAN TWS interface

```
...
//*****
//*          JCLLIB ORDER=(&XHLQ. . &NODE)
...

```

## Providing Access to JOB/SCAN for TWS

Follow the procedure in this topic to provide access to JOB/SCAN for TWS.

### Modify J00CTWS

The J00CTWS CLIST member will become a small repository for information pertaining to the TWS for z/OS configuration at your site. You will be asked to supply information such as TWS for z/OS subsystem names, high level qualifiers for both your TWS for z/OS product data set names (those allocated to your ISPF session) and your TWS for z/OS database data set names (those for the Application Database, JS file, and so on).

The information has been categorized into four separate and distinct areas and have been notated within J00CTWS as Modification Areas 1 through 4. The following instructions will direct you to each Modification Area to supply information needed for the interface.



ASG strongly recommends that you read through, and gather and jot down the requested information prior to beginning the edit of J00CTWS. This will greatly reduce the chance of errors.

### Modification 1

#### To add TWS subsystem data

1. In J00CTWS member, locate the code titled Begin Modification Area 1.
2. JOB/SCAN LIBRARIES: Replace `xxxxxxx .xxxx` with the high-level data set qualifier of the existing JOB/SCAN libraries. See note 1.

DJSNAME : \_\_\_\_\_



Use the same name as step 1 in [Method 3 Part A—Modify LIBDEFJS CLIST](#) .

3. TWS MESSAGE LIBRARIES: Replace `TWS .V8R2M0 .SEQQMSG0` with the complete data set name of the TWS message library if this value is not the library name you use. See note 2.

EQQMLIB : \_\_\_\_\_

4. TWS SUBSYSTEMS: Replace `TWS1 TWS2` with the names of subsystems with which the JOB/SCAN TWS for z/OS interface will interact. List as many as needed; JOB/SCAN uses the first subsystem listed as the default subsystem. See note 3.

SUBSYS\_LIST : \_\_\_\_\_

If you run TWS on more than one LPAR, you must specify the LPAR SMFID and TWS SUBSYS names in order to display the correct TWS subsystem name on panel JOOPMENU.

- a. Add these lines after the `EQQMLIB =` statement:

```

IF MVSVAR(SYSNAME)="AAAA" THEN SUBSYS_LIST = "1111"
ELSE
IF MVSVAR(SYSNAME)="BBBB" THEN SUBSYS_LIST = "2222"
ELSE SUBSYS_LIST = "3333"

```

where:

AAAA and BBBB are your LPAR SMFID names.

1111, 2222, and 3333 are the TWS SUBSYS names.

b. Remove (or comment out) the SUBSYS\_LIST statement. For example:

```
/* SUBSYS_LIST = "TWS1 TWS2" <=== add additional subsystems*/
```

5. Change the space attributes as necessary. See note 4.

### Modifications to JOOCTWS CLIST member Part 1

```

/*                BEGIN MODIFICATION AREA 1                */
1.  DJSNAME  = "XXXXXX.XXXX"          /* <=== change JOB/SCAN HLQ */
2.  EQQLIB   = "TWS.V8R2M0.SEQQMSG0" /* <=== change to Message Lib*/
3.  SUBSYS_LIST = "TWS1 TWS2" /* <=== add additional subsystems*/
4.  JLSCHR_INFO = "SPACE(1 1) TRACK VOLUME()" /* <=== chg if reqd.*/
/*                END MODIFICATION AREA 2                */

```

### Modification 2

In order to resolve variables in JCL that originates outside of the TWS for z/OS environment (for example, a user's private JCL library), the job must be associated with an application. If the user does not specify an application name, and the user also indicates they want JOB/SCAN to create a temporary application, then the values used to create that temporary application come from the fields in this part of the CLIST member.

#### To set up for resolving variables outside TWS for z/OS

1. In JOOCTWS member, locate the code titled Begin Modification Area 2.
2. Substitute the name of the default job for XXXXXX.

```
DEFAULT_JOB      : _____
```

3. Substitute the name of the default application for XXXXXXXX. Use a name that will NOT be used by the site as an application name.

```
DEFAULT_APPLICATION : _____
```

4. Substitute the name of the default workstation for XXXX. Use a value that is a valid workstation.

```
DEFAULT_WORK_STATION: _____
```

5. Substitute the number of the default operation for 001.

```
DEFAULT_OPERATION : _____
```

**Modifications to J00CTWS CLIST member Part 2**

```

/*              BEGIN MODIFICATION AREA 2              */
/*-----*/
/**** Set default variables for JTWS Edit Macro processing ****
A job must be associated with an application for TWS variables
to be correctly resolved. If Job/Scan is not able to locate the
correct application based on values entered on the Edit Macro
panel, then the following data values will be used to create
a unique temporary application(NF). Please ensure that the
values entered are valid for your TWS for z/OS installation. */
/*-----*/
DEFAULT_JOB          = "XXXXXXX" /* <-- CHANGE TO VALID JOB NAME */
DEFAULT_APPLICATION  = "XXXXXXX" /* <-- CHANGE TO UNIQUE APPL NAME */
DEFAULT_WORK_STATION = "XXXX"   /* <-- CHANGE TO VALID WORKSTATION*/
DEFAULT_OPERATION    = "001"    /* <-- CHANGE TO APPROPRIATE VALUE*/
/*              END MODIFICATION AREA 2              */

```



The JTWS Edit macro derives a temporary, uniquely-named application in the application database from the value entered in step 2 above when the user does not enter an application name on the JOB/SCAN for TWS Operation Selection Criteria panel. If this default value is a duplicate of an application already in the database, JCL validation cannot occur.

**Modification 3**

This area contains information describing the data set allocations under the EQQJBLIB ddname for each TWS for z/OS subsystem listed in Modification Area 1. In addition, if your site utilizes the EQQUX002 Exit Routine for a TWS for z/OS subsystem to retrieve JCL from a data set other than one listed under EQQJBLIB, you will be able to specify certain exit routine parameters in this area, allowing JOB/SCAN to retrieve, resolve and validate that JCL as well.

For each TWS for z/OS subsystem that you specified in Modification Area 1, you should have a corresponding "WHEN" statement identifying that subsystem followed by a "RETURN" statement containing the names of the data sets contained in the EQQJBLIB ddname for that subsystem.

**To describe data set allocations under EQQJBLIB**

1. In J00CTWS member, locate the code titled Begin Modification Area 3:
2. Change "TWS1" in the first WHEN statement to your default (primary) subsystem.

```
default subsys      : _____
```

3. Change "TWS.V8R2M0.JOBLIB" in the RETURN statement to the data set name(s) contained in the EQQJBLIB ddname for your default (primary) subsystem. Be sure to use proper REXX continuation syntax when listing more than one data set.

```
default subsys joblib:_____
```

4. The next WHEN/RETURN clause shows how to list additional subsystems with multiple data sets associated with the EQQJBLIB ddname. If your site has more than one subsystem, you may list as many WHEN/RETURN

clauses as necessary to describe your subsystem(s) EQQJBLIB ddname data sets. Likewise, if your site uses only one subsystem, you may delete any additional WHEN/RETURN clauses in this exec. This exec may be modified at any time in the future if your TWS for z/OS environment changes.

```
alternate subsys      : _____
```

### Modifications to J00CTWS CLIST member (Part 3A)

```

/*              BEGIN MODIFICATION AREA 3              */
/*Associate the PDS datasets allocated to the EQQJBLIB ddname
for each TWS z/OS subsystem Job/Scan will interface with.
DSN LIST:
SELECT              /* TWS EQQJBLIB concatenation      */
  WHEN SUBSYS="TWS1" THEN /* <=== change subsystem name      */
    RETURN "TWS.V8R2M0.JOBLIB" /* <=== change data set name      */

  WHEN SUBSYS="TWS2" THEN /* <=== change subsystem name      */
    RETURN "PROD.TWS2.CNTL" , /* <=== change data set name      */
        "TEST.TWS2.CNTL" , /* <=== change data set name      */
        "WORK.TWS2.CNTL" /* <=== change data set name      */
  OTHERWISE
    RETURN "NONE"
END
/*              END MODIFICATION AREA 3              */

```

To simulate a JOB-LIBRARY-READ exit (EQQUX002 Exit Routine), you will need to add additional qualifiers to the WHEN clause. These qualifiers are the Application ID, Job Name, Workstation Name and Operation Number. The below figure shows an example of how such an exit would be written in Modification Area 3.

### Modifications to J00CTWS CLIST member for subsystem list (Part 3B)

```

....
DSN_LIST:
SELECT              /* TWS EQQJBLIB concatenation      */
  WHEN SUBSYS="TWS1"
    & ADID = "XXXAPPL1",
    & JOBNAME = "BR14JOBA" ,
    & WSNAME = "SYS1",
    & OPNO = 1 THEN
    RETURN "TWS1.PRIVATE.JCLLIB"
....

```

## Modification 4

After modification, this area will contain information about server LU names and high level qualifiers of the data set names for both the TWS for z/OS product libraries and the database data sets for each subsystem you have listed.

### To set LU names

1. In J00CTWS member, locate the section titled Begin Modification Area 4.
2. Change "TWS1" in the first WHEN statement to your default (primary) subsystem.

```
default SUBSYS      : _____
```

3. The RETURN statement returns a string of positional parameters:

- **LUNAME** -- the first parameter is the LU name. Use `NOT-USED` if an LU name is not needed.
- **TWSPRODUCTHLQ** -- the second parameter is the TWS product data set qualifier.
- **TWSDATABASEHLQ** -- the third parameter is the TWS database data set qualifier.

Examples (each parameter is underlined):

```
RETURN "NOT-USED TWS.V8R2M0 TWS.V8R2M0.TWS1"
RETURN "NET.LUNAME2 TWS.V8R2M0 TWS.V8R2M0.TWS1"
```

Parameters for default SUBSYS: \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_

If your site has more than one subsystem, you may list as many WHEN/RETURN clauses as necessary to describe your subsystem(s) server LU names and data set qualifiers. Likewise, if your site uses only one subsystem, you may delete any additional WHEN/RETURN clauses in this exec. This exec may be modified at any time if your TWS for z/OS environment changes.

#### Modifications to JOOCTWS CLIST member for subsystem list

```
/*          BEGIN MODIFICATION AREA 4          */
/* Associate the LUNAME, the HLQ for the TWS z/OS data sets, */
/* and the HLQ of the TWS z/OS database data set names.      */
/* Do this for each TWS z/OS subsystem Job/Scan will         */
/* interface with.                                          */
/*
SUBSYS_DETAIL:
SELECT_          /* TWS subsystem detail          */
WHEN SUBSYS="TWS1" THEN /* <=== change subsystem name */
  RETURN "NOT-USED TWS.V8R2M0 TWS.V8R2M0" /* <=== CHANGE */

WHEN SUBSYS="TWS2" THEN /* <=== change subsystem name*/
  RETURN "LUNAME TWSPRODUCTHLQ TWSDATABASEHLQ" /*<=== CHANGE*/
          /* <=== add additional WHEN stmts.*/

OTHERWISE
  RETURN "NONE"
END
/*          END MODIFICATION AREA 4          */
```

4. Save the member.

#### Setting the JCL Edit Tool in TWS

The Edit tool can be set by each user or can be set globally. Choose the appropriate option for your site:

#### Setting the JCL Edit Tool for a Single User

Tivoli Workload Scheduler for z/OS includes a plug-in point for a JCL edit tool. It is displayed when the edit/browse JCL function is invoked by the AD dialogs. Users can specify their own tool. JOB/SCAN includes a panel for this purpose.

From the Tivoli Workload Scheduler for z/OS main menu go to option 0.6. Then, change the `PANEL NAME` to `J00PTS00` as shown in the below figure.



Each Tivoli Workload Scheduler for z/OS user must set the JCL Edit Tool Information.

## Setting the JCL Edit/View tool

```

----- SETTING JCL EDIT TOOL INFORMATION -----
Command ==>

Enter/change data below:

PANEL NAME      ==> J00PTS00      Specify the name of the panel to
                                be displayed when editing JCL from
                                AD data base.

                                This panel name is used to provide
                                a way to call a user tool to edit JCL.

                                Default value is EQQAJCLE which is
                                a dummy panel.

```



If you do not have a JCL tool installed in TWS option 0.6, the `PANEL NAME` field names `EQQAJCLE` as the Edit tool; if you already have another JCL Tool installed, then the `PANEL NAME` field names a different panel. In this case, JOB/SCAN for TWS interface includes a feature for preserving access to the previously installed tool.

See [Preserving Access to an Existing JCL Edit Tool](#) for instructions to preserve access to the previously installed editing tool.

## Setting the JCL Edit Tool Globally for all Users



In the future, when upgrading or installing maintenance to IBM's Tivoli product, be sure to apply these changes to a new copy of the updated panel.

### To set up the Edit tool for all users

1. Copy `EQQOPCAP` from the TWS for z/OS panel library to the JOB/SCAN panel library.
2. Add the following statements to the PROC section of `EQQOPCAP`:

```

&XJCLPAN = 'J00PTS00'      /*JOB/SCAN JCL edit tool panel*/
&XMANPAN  = 'MYPANEL'      /*name of additional edit tool*/
&XMANOPT  = 'MY EDIT'      /*option name for edit tool */
&XMANTXT  = 'My JCL Editor' /*Description for edit tool */
VPUT (XJCLPAN XMANPAN XMANOPT XMANTXT) PROFILE

```

3. Substitute your own quoted strings according to the descriptions in this table.

Field	Description
XJCLPAN	Use JOB/SCAN's EDIT TOOL PANEL ID, 'J00PTS00'
XMANPAN	Use your own panel id. Write it : _____
XMANOPT	The name of your own edit tool. Write it: _____

Field	Description
XMANTXT	Your own description of the edit tool. Write it: _____

**Provide JOB/SCAN Users Access to TWS Application and Plan Validation Panels**

Add the new user function to the JOB/SCAN Main Menu by modifying the JSOPT01 member in the JOB/SCAN SJJSPARM library. An example of this is shown in the below figure. The command for the feature, here shown as option S (the value for the SELECT keyword), is determined by the customer. You may also change the function description (the value for the DESCRIP keyword).

**Modifications to the JSOPT01 member**

```

***** Top of Data *****
** JSOPT01 ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** *
*
SYSTEM IDENTIFICATION
  TITLE1='WIDGET MANUFACTURING INC'
ACCESS CONTROL
  INITIALIZATION      =ALL
  JOB/SCAN            =ALL,EXIT=ANY,PARM=
  SELECT LISTS       =ALL
  OPTIONS             =ALL
  ADMINISTRATOR       =ALL
  JOB/SCAN USAGE REPORT =ALL
1.  USER FUNCTION    =CMD=J00CTS20,
1.  SELECT=S,
1.  NAME=,
1.  DESCRIP='Validate TWS for z/OS Scheduler JCL'

----- ASG-JOB/SCAN PRIMARY MENU ----- SELECTION ==>
0 - Hardcopy Output and JOB Statement Definition   USERID - SYSUSR
1 - Validate JCL                                  PREFIX  - SYSUSR
2 - Change / Reformat JCL                        DATE    - 2005-03-23
3 - Processing Options                           TIME    - 15:20
4 - Selection List Maintenance                   APPLID  - JS
5 - Assign Alternate Options Members
6 - JOB/SCAN Usage Report
S - Validate TWS for z/OS Scheduler JCL
X - Exit

```

**Activate the JVAL command**

JOB/SCAN for TWS adds a new command to Tivoli Workload Scheduler for z/OS called JVAL. This allows JCL validation with JOB/SCAN from any panel within TWS. Make the JVAL command available to users by using J00CCMDT CLIST to copy and update the TWS ISPF command table.





In the future, when upgrading or installing maintenance to IBM's Tivoli product, be sure to apply these changes to a new copy of the updated TWS ISPF command table.

## Command Syntax

```
J00CCMDT ADD <table-input-library> <table> <table-output-library>
```

To make the change furnish the input library name, table name and output library name; the output library, if it already exists must be a PDS with LRECL 80.

Write down:

```
table input library _____
table name          _____
table output library _____
```

Example:

```
TSO J00CCMDT ADD TWS.V8R2M0.SEQQTBL0 EQQACMDS NEW.TABLE.ISPTLIB
                  (input data set)      (output data set)
```



- If you created a new output table library, allocate it ahead of the TWS table library.
- Do not use quotes or the square brackets [ ] in the command.

## Provide Access to the Interface Features from TWS

### To include the JOB/SCAN ISPF data sets in ISPSLIB and ISPTLIB

1. Modify your current method of accessing TWS (LOGON procs or CLISTs).
2. Upon completion of this modification, log off and then back on to TSO/ISPF.

### Allocating the EQQCLIB DD for TWS Distributed Environments

**To verify JCL when the TWS for z/OS controller is configured to support a distributed TWS environment (also known as End-to-End or E2E)**

1. Edit the J00STS2B member in the JOB/SCAN skeleton library.
2. Locate step DPTRIAL.
3. Uncomment the EQQCLIB DD:

### Setting up the End-to-End processing

```
...
//DPTRIAL EXEC PGM=EQQBATCH,
//          REGION=4096K,
//          PARM='EQQDTTOP',
//          COND=(4,LT)
...
//* UNCOMMENT EQQCLIB DD FOR END TO END SYSTEMS
//*EQQCLIB DD DISP=SHR,DSN=&XTWSHLQ..SCRPTLIB
...
```

4. Save the member.

## Activate TWS Scheduler Interface

Do all the parts of this step from [Set the SCHEDULER Run Option](#) through [Background Processing from JOB/SCAN Panels](#).

### Set the SCHEDULER Run Option

Set the SCHEDULER Run Option to TWS in the SJJSPARM library member JSOPT02:

```
RUN OPTIONS
...
  SCHEDULER=(TWS, subsystemname)
```

where `subsystemname` is a two- to four-character string naming the TWS subsystem.

### Turn on the Scheduler Processing Option

#### To turn on scheduler processing

1. Choose Processing options from the JOB/SCAN. (Alternately, select Processing Options from the ASG-JOB/SCAN Primary Menu, if available.)

#### Selection Processing Options

```
----- JOB/SCAN - JCL VALIDATION -----
COMMAND ===> P

      F - Foreground execution mode          P - Processing Options
      B - Batch execution mode              E - Edit JCL member
                                          BR - Browse JCL member

Input Library  ===> SYSUSR.JCL
Input Type    ===> J                      (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ===> PDS                    (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ===> BR14
...

```

2. JOB/SCAN displays the JOB/SCAN PROCESSING OPTIONS panel.
3. Type a Y in the Use SCHEDULER interface field. Press PF3 to return and accept this value.

#### Processing Options Panel

```
D02PEP ----- JOB/SCAN PROCESSING OPTIONS -----
COMMAND ===>

Use Error Table           ===> N          (Y/N)
Condition Code Processing ===> N          (Y/N)
Catalog Access           ===> Y          (Y/N)
VTOC Access              ===> Y          (Y/N)
PDS Member Verification  ===> Y          (Y/N)
Security Exit DSNAME Check ===> Y          (Y/N)
Program Checking         ===> Y          (Y/N)
Disable SMS Processing   ===> N          (Y/N)
IDCAMS Processing       ===> Y          (Y/N)
Use SCHEDULER interface  ===> Y          (Y/N)
Job Limit                ===>           (1 to 999 or Blank For No Limit)
Access Control Cards     ===> Y          (Y/N)
...

```

### Activate DDs in FILE ALLOCATIONS Statement

Make sure that the attributes for the JLSCHR DD in the FILE ALLOCATIONS statement in JSOPT02 are suitable for your site.

```
JLSCHR =DSN=&&JLSCHR,NEW,DELETE, SCHED WORK FILE
      UNIT=SYSDA, VOLSER=OPTIONAL, ...
      TRKS,PRIM=2,SEC=1,RECFM=FB, ...
      LRECL=80,BLKSIZE=27920 ...
```

### Control Table CE: Scheduler Restart Program Names

Control Table CE contains the names of scheduler restart programs. JOB/SCAN will simulate the operation of the restart program when one of these programs is found.

### Conditions Requiring CE Entries

Control Table CE entries are required if the customer site is using a scheduling product.

### Format

Enter Control Table CE data as follows:

Column	Entry
6-7	CE
8-15	Scheduler restart program name

### Example

```
.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....
CEU11RMS
CECTRCTR
CEEQQDELDS
```

### Add DD Statements to Batch Jobs

If you intend to run the JOB/SCAN TWS interface in batch, add these DD statements to the JCL running JS6PROC:

```
//EQQMSG DD SYSOUT=*
//EQQMLOG DD SYSOUT=*
//EQQDUMP DD SYSOUT=*
//EQQMLIB DD DISP=SHR,DSN=TWS.XXXXXXX.SEQQMSG0 <== Modify
//STEP01.JLSCHO DD SYSOUT=*
//STEP01.JLSCH DD *
```

For DD JLSCH, this table lists the control card values (if any value contains delimiters or spaces place quotes around it):

Control Card	Description
ADID(applicationID)	The application description ID name.
WSN(workstationname)	The workstation name.
OPRNUM(operationnumber)	The operation number.
IADATE(arrivaldate)	The input arrival date.
IATIME(arrivaltime)	The input arrival time.

Control Card	Description
JCLVTABL (tablename)	The TWS variable table to use.

Precede the first card with TWS. Place a continuation character (-) at the ends of lines that continue to the next line:

```
//STEP01.JLSCH DD *
TWS JOBNAME ('MY.JOB') -
ADID (THEAPPL)
```

### Background Processing from JOB/SCAN Panels

If you intend to run the JOB/SCAN TWS interface in background mode from JOB/SCAN panels, add these DD statements directly to JS6PROC:

```
//JLSCHO DD SYSOUT=*
//JLSCH DD DSN=MY.CNTL.CARD,DISP=SHR <== Modify
//EQQEMSG DD SYSOUT=*
//EQQMLOG DD SYSOUT=*
//EQQDUMP DD SYSOUT=*
//EQQMLIB DD DISP=SHR,DSN=TWS.XXXXXXX.SEQQMSG0 <== Modify
```

The JLSCH DD should refer to a data set containing the control cards as described in [Add DD Statements to Batch Jobs](#).

## Reporting on TWS Special Resources (Optional)

JOB/SCAN supplies a load library member and its COBOL source, called TWSSTD1. JOB/SCAN uses the module to query TWS (through TWSSPRES) about a missing data set. If TWS reports that it is waiting for the data set, a special message is inserted into the SJL reporting this fact.

In particular, TWSSTD1 changes the JOB/SCAN message:

from:

```
//DD1 DD DSN=SYSUSR.SPECIAL.RESOURCE,DISP=SHR
***ERROR - DSS4050E - DATA SET NOT FOUND IN CATALOG
*ADVISORY - DSS8900A - DSN = "SYSUSR.SPECIAL.RESOURCE "
```

to:

```
//DD1 DD DSN=SYSUSR.SPECIAL.RESOURCE,DISP=SHR
**WARNING - DSS9101W - TWS WILL WAIT FOR SPECIAL RESOURCE.
*ADVISORY - DSS9102A - TWS RESNAME = "SYSUSR.SPECIAL.RESOURCE"
```

for datasets identified as TWS special resources.

Otherwise, JOB/SCAN reports the missing data set is reported as a JCL error.

### To use the program in batch

1. The load modules TWSSTD1 and TWSSPRES must be loaded from an APF authorized library.



TWSSTD1 cannot be used with foreground or Edit macro validation if JOB/SCAN is set up with LIBDEFs (unless the LOAD library is preallocated through another means).

- JOB/SCAN must be running as an APF-authorized program (see instructions in [Running the Product in an APF-authorized Environment](#) on how to run JOB/SCAN APF-authorized in batch. The JOB/SCAN load library must be APF authorized, and J02YJSA must be renamed to J02YJS to make J02YJS an APF authorized module.)
- JOB/SCAN's JS6PROC must have the following DD statements:

```
//EQQEMSG DD SYSOUT=*
//EQQMLOG DD SYSOUT=*
//EQQDUMP DD SYSOUT=*
//EQQMLIB DD DISP=SHR,DSN=TWS.XXXXXXX.SEQQMSG0 <== Modify
```

If these DDs are missing TWSSTD1 will be called, but the results will appear as if the standards program was never called or not working, and no error conditions will be reported.

- If TWSSTD1 is to be called as a standards program specify the TWS subsystem name as a user parameter on the same statement. The optional LUNAME may be coded after the subsystem name with a comma between. In SJJSPARM library member JSOPT02 specify these two lines in the Run Option section:

```
RUN OPTIONS
  STANDARDS NAME=TWSSTD1, 'TWSX, LUNAME.XXXXXXX'
  COBOL EXIT ENVIRONMENT=COBOL/370
```

Other equivalent values for the COBOL EXIT ENVIRONMENT statement are COBOL/II or COBOL/MVS. For more on Run Options see [Reference Information](#).

## Verify Installation

To verify the installation of the interface, you need a TSO UserID with authority to access JOB/SCAN and the TWS for z/OS ISPF dialogs.

### Verify the JTWS EDIT Macro

#### **To confirm that the EDIT macro component of the interface works**

- Edit a member in ISPF containing TWS JCL variables that resides in a non-TWS for z/OS data set (for example, a private JCL library).
- Type `JTWS` on the command line.
- Press Enter.
- The first time through, each user should get the Advisory panel. Type Y on the command line to continue.
- JOB/SCAN displays the JOB/SCAN for TWS Operation Selection Criteria panel. The TWS for z/OS default (primary) subsystem name and the JCL job name should appear in the appropriate fields. The "CURRENT PLAN ENDS" field should display the end date of the Current Plan on the selected subsystem. Enter NO in the CP and AD search fields. Enter YES in the PROVIDE NF ENTRY field.
- Press Enter.
- JOB/SCAN displays the JOB/SCAN for TWS Operation List panel. The default application ID specified during the install should appear as the only selectable item. Place an asterisk (\*) in the Variable Table field and select the default application listed.

8. Press Enter.
9. If a member has promptable variables, JOB/SCAN displays the Promptable Variables panel. Make any necessary changes.
10. Press Enter. If the member does not contain promptable variables, go to Step 7.
11. JOB/SCAN displays the Variable Table List. Select the appropriate entry.
12. Press Enter to continue processing; press PF3 to return.
13. The Structured JCL Listing will be the next screen displayed. Verify that variables were resolved.

### Verify ASG-JOB/SCAN Primary Menu (optional)

#### ***To confirm selection of the Validate TWS for z/OS Scheduler JCL option on the ASG-JOB/SCAN Primary Menu***

1. Start JOB/SCAN product.
2. Choose the Validate TWS for z/OS Scheduler JCL option from the ASG-JOB/SCAN Primary Menu.
3. Verify that JOB/SCAN displays the JOB/SCAN FOR TWS - JCL VALIDATION panel.

### Verify JVAL Command

#### ***To confirm the availability of the JVAL command, Test 1***

1. Select the TWS for z/OS ISPF product dialog.
2. TWS for z/OS displays its OPERATION PLANNING AND CONTROL menu. Type JVAL on the command line.
3. Press Enter.
4. Verify that JOB/SCAN displays the JOB/SCAN FOR TWS -- JCL VALIDATION panel.

#### ***To confirm the availability of the JVAL command, Test 2***

1. From within the TWS for z/OS ISPF dialog, type JVAL 2 on the command line.
2. Press Enter.
3. Verify that JOB/SCAN displays the JOB/SCAN FOR TWS -- CURRENT PLAN VALIDATION panel. Press PF3 or End.
4. Type JVAL 3 on the command line.
5. Press Enter.
6. Verify that JOB/SCAN displays the JOB/SCAN FOR TWS - TRIAL PLAN VALIDATION panel.
7. Press PF3 or End.
8. Type JVAL 1 on the command line.
9. Press Enter.
10. Verify that JOB/SCAN displays the JOB/SCAN FOR TWS - VALIDATE APPLICATION panel. Press PF3 or End.

## Verify the Skeleton Library

### *To confirm the availability of the skeleton library*

1. In the JOB/SCAN FOR TWS -- CURRENT PLAN VALIDATION panel, blank out JOBNAME, APPLICATION ID, OWNER, AUTHORITY and WORKSTATION fields.
2. Type E on the command line.
3. Press Enter.
4. JOB/SCAN displays a JCL member containing your default jobcard and a product validation and copyright notice.

## Verify the Edit tool

### *To confirm that the Edit tool is properly available while Browsing or Editing an application*

1. Select an application within TWS for z/OS application database to browse or edit. Use the "J" line command on a JOB in the application.
2. Upon entering the "J" line command, you should be presented with the JOB/SCAN FOR TWS - VIEW JCL panel. The panel is context sensitive and will say VIEW or EDIT depending upon the mode TWS for z/OS was in (i.e. "J" line command to Browse or Edit the JCL).
3. Select Option 1. JOB/SCAN will retrieve the JCL from the EQQJBLIB ddname data sets as specified during the installation. The JCL will be displayed in a VIEW or EDIT mode where the JTWS Edit Macro may be used for variable resolution and JCL validation.
4. Press PF3 twice to return to the TWS for z/OS panel.

## Verify Validation from JOB/SCAN Panels

### *To confirm that Validation from JOB/SCAN panels resolves TWS*

1. Make sure that all the steps in [Activate TWS Scheduler Interface](#) are completed.
2. Start JOB/SCAN product.
3. Select Validate JCL on the ASG-JOB/SCAN Primary Menu panel.
4. Fill in fields for Input Library, member and type JOB/SCAN JCL Validation panel.
5. Use the F command to start validation.
6. Type Y on the command line of the Advisory panel.
7. Press Enter.
8. Fill in the fields on the Operation Selection Criteria panel.
9. Press Enter.
10. Select an application/operation, verify and change the fields on the Operation List panel.
11. Press Enter.
12. If the job contained promptable variables, JOB/SCAN displays the Promptable Variables panel. Make any necessary changes.
13. Press Enter.

14. JOB/SCAN displays the SJL. Check the SJL to verify that the process worked as expected.
15. Press PF3 to return to the JOB/SCAN JCL Validation panel.

### Verify Single Job Validation in Batch

#### To confirm the single job validation in batch

1. Follow the instructions in [Validate a Single Member in Batch](#) to verify single job validation through batch. Use the JCL sample from the below figure for this verification.

#### Modifying batch JCL to validate a single member

```
//MYJOB    JOB (ACCT), 'PROGRAMMER NAME'
//*****
//* EXECUTE JOB/SCAN (MVS SYSTEMS)
//* REF:  SAMPLIB (JSMVS)
//
//* NOTE: MODIFY JOB STATEMENT AND ITEMS INDICATED BY ARROWS <==
//*****
//JSSTEP1 EXEC JS6PROC, PRM=S
//SYSIN   DD   *
INPUT SELECTION
    DSN=YOUR.JCLLIB           <== INPUT JCL LIBRARY
    SELECT=YOURJOB            <== MEMBER TO SCAN
    SOURCE TYPE=PDS           <== TYPE OF INPUT LIB
REPORT OPTIONS
    OPTIONS IN EFFECT        =PRINT           <== PRINT OPE REPORT?
    STRUCTURED JCL LISTING=PRINT             <== PRINT SJL REPORT?
//STEP01.JLSCHO DD SYSOUT=*
//STEP01.JLSCH  DD   *
          TWS ADID (MYAPPL) OPRNUM (003) IADATE (100325) IATIME (1200)
```

2. Verify that in the SJL produced the TWS variables are resolved.

## JCL Parameters for the JVAL Command JCL

These are the parameters for the JCL generated by the panels launched by the JVAL command. See the generated JCL for additional details.

Parameter	Description	Current Plan	Trial Plan	Application
SUBSYS	Name of the TWS Controller subsystem.	x	x	x
DEPENDENCIES	Internal, used to ensure that JOB/SCAN processes JCL taking Predecessors and Successors into account.	x	-	-
JOBNAME	A job name or mask to limit the operations to be validated. The wildcard characters % and * may be used.	x		



Parameter	Description	Current Plan	Trial Plan	Application
ADID	An Application Description ID name or mask. The wildcard characters % and * may be used. If the following field has a value this field will define the <i>lower</i> end of a range of Applications.	x		x <sup>a</sup>
ADID2	An Application Description ID name or mask. The wildcard characters % and * may be used. If this field has a value it will define the <i>upper</i> end of a range of Applications and the preceding field will define the lower end.	x	x	
OWNER	An Owner ID name or blank. The wildcard characters % and * may be used. This field can be used to limit the Applications examined.	x	x	
GROUP	An Authority Group ID or blank. The wildcard characters % and * may be used. This field can be used to limit the Applications examined.	x	x	
WSNAME	A Work Station name or blank. The wildcard characters % and * may be used. This field can be used to limit the Applications examined.	x	-	
PRIORITY	A number or blank. Only Applications with this priority or greater will be examined.	x	x	
STATUS	<p>A string of status codes. Use to validate jobs of the specified status code. You can specify one or more status codes on the input field. If blank, status codes ARW* will be used. If all are desired, all must be specified. The following are valid status codes:</p> <ul style="list-style-type: none"> <li>• A - Ready. No predecessor defined.</li> <li>• C - Complete.</li> <li>• D - Deleted.</li> <li>• E - Operation ended in error.</li> <li>• I - Interrupted operation.</li> <li>• R - Ready. All predecessors are complete.</li> </ul>	x	x	

<sup>a</sup>When specifying an application ID for application validation, you may not use a mask. Use the full ID.

Parameter	Description	Current Plan	Trial Plan	Application
	<ul style="list-style-type: none"> <li>• S - Started.</li> <li>• U - Undecided. The status is not known.</li> <li>• W - Waiting for a predecessor to complete.</li> <li>• * - Ready. At least one predecessor was non-reporting. All predecessors are complete.</li> </ul>			
IA (FROM DATE)	<i>From</i> date and time. This date field in the format yymmddtttt defines the lower end of the range of dates to scan. It must be less than or equal to the Current Plan end date.	x		
IA (TO DATE)	<i>To</i> date and time. This date field in the format yymmddtttt defines the upper end of the range of dates to scan. It must be greater than or equal to the From date.	x		
GROUPDEF	<i>A group definition</i> or blank. Use to validate jobs that are in specific application groups. Specify the name of the application group. Wildcards (* and %) are allowed.	x	x	
WSTWS	Internal. Filters out Fault Tolerant Work Stations as these are normally for Distributed systems.	x	-	x
ADIA	The date and time that will be used to validate the Application.		-	x
VALTO	<i>From</i> date. This date field in the format yymmdd defines the lower end of the Trial Plan to validate. It must be greater than or equal to the Current Plan end date.		x	
VALFROM	<i>To</i> date. This date field in the format yymmdd defines the upper end of the Trial Plan to validate. It must be greater than or equal to the From date, and cannot be more than 21 days beyond the From date.		x	

## Preserving Access to an Existing JCL Edit Tool

The information in this topic is only used if an individual user needs to make these changes. Generally, the information in [Setting the JCL Edit Tool in TWS](#) makes this available to all users.

### **To preserve and provide access to a previously installed JCL Edit tool**

1. Invoke the JOB/SCAN JCL edit tool.

Browse the application list using TWS option 1.4.1. Then select an application. Enter the OPER command to list the operations. Against any of the JCL operations, enter the J command. This displays the JOB/SCAN FOR TWS-Edit/View JCL panel.

#### JOB/SCAN for TWS JCL Edit /View JCL Tool

```

----- JOB/SCAN FOR TWS - VIEW JCL -----
Option ==>
Subsystem Name: TWSC
Application   : GLDLYPOST
Workstation  : CPU1
Operation No : 20
Job Name     : PJGL0020

0 OPTIONS    - Specify Alternate JCL Edit TOOL
1 VIEW      - VIEW JCL from EQQJBLIB
X EXIT
NOTE        : From VIEW use command JTWS to validate PJGL0020 with resolved
              variables using JOB/SCAN from ASG, Inc.

```

2. Add the previously installed JCL edit tool using option 0.
3. Enter option 0 and set the details of the pre-existing JCL edit tool. Enter the PANEL NAME of your alternate JCL tool. Then enter the OPTION and OPTION TEXT fields.

#### Specifying an alternate JCL editing/viewing tool

```

----- JOB/SCAN FOR TWS - ALTERNATE JCL EDITING TOOL -----
Command ==>
Enter/Change data below: Enter END to save or CANCEL to exit without saving

PANEL NAME    ==> MYPANEL    Specify the name of the panel to be displayed
                               when editing JCL from
                               within the TWS plug-in.
OPTION NAME   ==> MY EDIT    Specify the option name to appear
                               on the plug-in menu.
OPTION TEXT   ==> My JCL Editor
                               Specify the description to appear
                               on the plug-in menu.

```

JOB/SCAN displays these extra fields on the JOB/SCAN panel.

#### JOB/SCAN Edit/View tool with alternate Edit/View tool installed

```

----- JOB/SCAN FOR TWS - VIEW JCL -----
Option ==>

Subsystem Name: TWSC
Application   : GLDLYPOST
Workstation  : CPU1
Operation No : 20
Job Name     : PJGL0020

0 OPTIONS    - Specify Alternate JCL Edit TOOL
1 VIEW      - VIEW JCL from EQQJBLIB
2 MY EDIT   - My JCL Editor
X EXIT

NOTE        : From VIEW use command JTWS to validate PJGL0020
              with resolved variables using JOB/SCAN from
              ASG, Inc.

```

## Reference Information

This documentation provides complete information for the ASG-JOB/SCAN (herein called JOB/SCAN) Administrator and user. The documentation describes how to use control statements, File Allocation statements, and Control Tables, explains the numerous Run and Report Option statements, and describes the numerous third-party interfaces.

Topic	Description
<a href="#">Run Option Statements</a>	Details the JOB/SCAN Run Option statements.
<a href="#">Control Tables</a>	Details the JOB/SCAN Control Table statements.
<a href="#">Scheduler Interfaces</a>	Outlines how JOB/SCAN interfaces with scheduler packages.
<a href="#">Security Interfaces</a>	Outlines how JOB/SCAN interfaces with security packages.
<a href="#">Other Interfaces</a>	Provides detailed information about the installation of the other interfaces that are compatible with JOB/SCAN.
<a href="#">Extended Parameter List Samples</a>	Provides samples of the contents of each data element in the Extended Parameter List.
<a href="#">Program Name Exit</a>	Provides detailed information about program name exit.
<a href="#">User Abends Issued by JOB/SCAN</a>	Lists user abend codes, the related modules, and explanations.
<a href="#">Selection Exit Interface</a>	Provides detailed information about selection exit interface.

---

# Run Option Statements

This section describes the elements comprising JCL statements and parameters. Telling JOB/SCAN how to deal effectively with JCL set of elements is exacting, but rarely needs to be done more than once, with occasional maintenance afterwards. Run Option statements control how JOB/SCAN processes the JCL and customize various reports and third-party interfaces. Run Option statements reside in members of the SJJSPARM library.

- Access Control Cards (ACC)
- Catalog Access (CAC)
- CONTROL-M Exits (CMX)
- CONTROL-M Variables (CMV)
- DBCS
- Default Unit Names (DUN)
- Device Codes (DVC)
- DMS Volume Serial (DMSV)
- Exec Parm String (EPS)
- GDG Processing (GDGP)
- Group ID (GRP)
- Headings (HDG)
- HSM Volume Serial (HSMV)
- ISPF Specifications (ISS)
- JCL Change Parameters (JCLCP)
- JES2 Command Character (J2C)
- JLJPA PROCLIB Type (JPAPT)
- JLJPC PROCLIB Type (JPCPT)
- Job Limit
- JOB/SCAN PROC (JSP)
- LIBRARIAN (LIBR)
- LINECNT
- Migrate Device Type (MDTP)
- Operating System Level (OSL)
- Output Specifications (OSS)
- Override Indicator (ORI)
- Prefix (PFX)
- Processing Specifications (PSS)
- Program Product Interface (PPI)

- Report Heading
- Resolve System Symbols (RSS)
- Scan JCL Only (SJO)
- Scheduler (SCHED)
- Security Exit
- SEMRESET
- Separator (SEP)
- Short Descriptions (SDS)
- SMS
- Span Members (SMB)
- Standards Name (STN)
- Subsystem (SSS)
- Suppress Error Messages (SEM)
- Updated JCL File Type (UJFT)
- Use Error Table (UET)
- Use PROC Exclude Table (UPCX)
- User ID (UID)
- Write Selected JCL (WSJ)

## Access Control Cards (ACC)

This Run Option directs JOB/SCAN to read members within the job stream that contain control cards, then print the control cards in the Structured JCL Listing. The libraries containing these members are designated in Control Table AB for ddnames or Control Table AQ for dsnames. A dataset name in a CG Control Table entry is not considered a Control Card dataset no matter what is coded in the AB, AD, or AQ Control Tables and will not be expanded nor verified. JOB/SCAN ISPF users can enter these values on the Processing Options panel rather than by coding this Run Option. Values entered on the panel override corresponding parameters of this Run Option coded in the SJJSPARM library.

### Format

```
▶▶ ACC= [YES, LIMIT=nn] ▶▶
      [NO]
```

### Example

```
ACCESS CONTROL CARDS=YES, LIMIT=12
```

## Parameters

Parameter	Default	Function
YES/NO	YES	Includes or excludes control cards in the Structured JCL Listing. These cards are also passed through to the JOB/SCAN Standards program. If <i>NO</i> is coded, control cards are excluded from the Structured JCL Listing, and are not passed to the Standards program. In the case of IDCAMS and DB2, the statements are not validated.
LIMIT= <i>nn</i>	99 [batch] No Limit [ISPF]	Designates the number of control cards printed from each member in the library accessed. To print all control cards, code LIMIT=0.

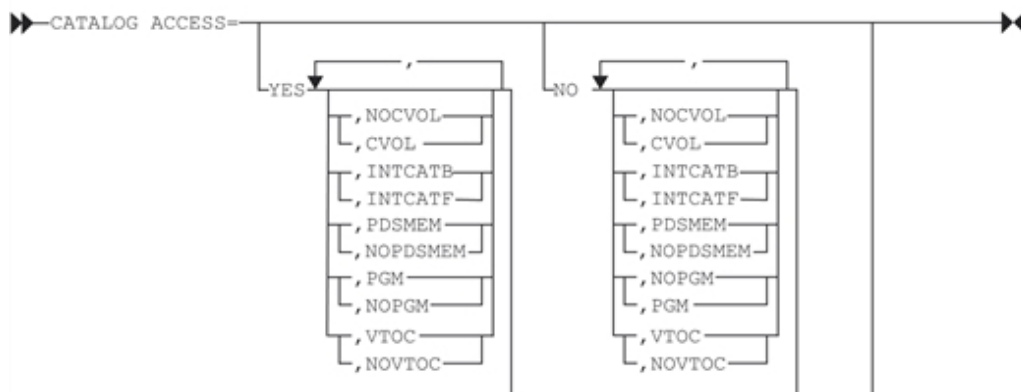


LIMIT does not affect the number of control cards read during DB2 and IDCAMS processing. It only affects the control statements that are printed in the Structured JCL Listing.

## Catalog Access (CAC)

This Run Option controls access to the system catalog and, in the case of disk datasets, the VTOCs. The parameters coded in the statement determine whether or not JOB/SCAN reads the catalogs and VTOCs. JOB/SCAN ISPF users can enter these values on the Processing Options panel rather than by coding this Run Option. Values entered on the panel override corresponding parameters of this Run Option coded in the SJJSPARM library.


### Format




### Example

```
CATALOG ACCESS=NO,NOVTOC,PGM
CAC =YES
```

## Parameters

Parameter	Default	Function
YES/NO	YES	Specifies whether or not to allow access to the system catalog. This parameter is mandatory and must be first in the list.
VTOC/ NOVTOC	VTOC	Specifies whether or not the DASD VTOCs are to be accessed for dataset information.
PGM/ NOPGM	PGM	Specifies whether or not to perform program validation. Default is PGM when CAC=YES,VTOC is coded; NOPGM is the default when CAC=NO is coded.
<div style="border: 1px solid purple; padding: 5px;">  PGM should not be specified unless program validation is required without normal catalog or VTOC access. Coding PGM results in a volume mount if a JOBLIB or STEPLIB dataset is not online. </div>		
PDSMEM/ NOPDSMEM	PDSMEM	Specifies whether or not JOB/SCAN is to validate partitioned dataset member names. Verification is done when a DSN=dsname(membername) parameter is coded in the JCL DD statement being validated.
CVOL/ NOCVOL	NOCVOL	Code CVOL if there are generation data groups (GDGs) cataloged on CVOL catalogs. Code NOCVOL if all GDGs are cataloged on ICF or VSAM catalogs.
INTCATB/ INTCATF	INTCATB	Specifies which direction an internal copy of the system catalog is searched. INTCATB designates a backward search of the catalog accessing newer entries first. INTCATF designates a forward search, accessing older entries first. When CAC=NO,NOVTOC is specified, device types are determined from the entries in Control Table AA.

 When CAC=NO,NOVTOC is specified, device types are determined from the entries in Control Table AA.

## CONTROL-M Exits (CMX)

This Run Option can be specified to invoke the CONTROL-M Exit 2 and Security Exit 2. These exits perform these functions:

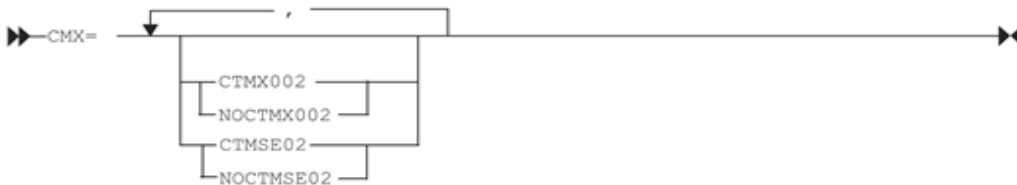
- Exit2 (CTMX002) is the CONTROL-M Submission Exit and allows each line submitted to be modified, deleted, or replaced. It can also be used to add /\*PRIORITY statements for production jobs.



- Security Exit 2 (CTMSE02) is related to this exit, and verifies that the user has the proper authority to read the JCL library.

If the CONTROL-M EXITS Run Option is specified, make sure that you have provided a DD statement in your JS6PROC that concatenates the library containing these exits to the JOB/SCAN libraries (JLLOD and SJJSLMOD if running under ISPF). If the CONTROL-M EXITS Run Option is not coded, the default is to select both exits. For more information about these exits, refer to the *CONTROL-M Installation and Security Guides*.

### Format



### Example

```
CONTROL-M EXITS=(CTMX002,CTMSE02)
```

### Parameters

Parameter	Default	Function
CTMX002/ NOCTMX002	CTMX002	Determines whether CONTROL-M EXIT 2 is to be invoked (to modify, delete, or replace each line submitted).
CTMSE02/ NOCTMSE02	CTMSE02	Determines whether CONTROL-M SECURITY EXIT 2 is to be invoked (to verify that the user has the proper authority to read the JCL library).

## CONTROL-M Variables (CMV)

This Run Option determines how CONTROL-M should respond to unresolved %% variables found in the input JCL stream. The NORESOLVE option specifies that CONTROL-M should ignore the unresolved variables and continue processing. RESOLVE specifies that CONTROL-M should stop processing and issue an error message when these variables are unresolved.

### Format



## Example

```
CONTROL-M VARIABLES=RESOLVE
```

## Parameters

Parameter	Default	Function
RESOLVE/ NORESOLVE	NORESOLVE	<p>Determines how CONTROL-M should respond when %% variables are unresolved.</p> <p>Specify RESOLVE to cause CONTROL-M to stop processing and issue an error message when these variables are unresolved.</p> <p>Specify NORESOLVE to cause CONTROL-M to continue processing when these variables are unresolved.</p>

## DBCS

This Run Option determines whether JOB/SCAN should support the IBM 5550 Terminal using the Double-Byte Character Set (DBCS). When the DBCS Run Option is set to YES, ISPF Edit will treat the data as mixed-mode DBCS data. When set to NO, ISPF Edit will treat the data as EBCDIC (single-byte data).



This option is provided to support the IBM 5550 Terminal using the Double-Byte Character Set.

## Format



## Example

```
DBCS=YES
```

## Parameters

Parameter	Default	Function
YES/NO	NO	<p>Determines whether JOB/SCAN should support the Double-Byte Character Set.</p> <p>Specify YES to support the IBM 5550 Terminal using the Double-Byte Character Set (DBCS).</p>

## Default Unit Names (DUN)

This Run Option assigns unit names to DASD, mass storage, reel tape, and cartridge tape devices for cataloged datasets in cases where JOB/SCAN is running on a system without IBM module IEFEB4UV available to obtain the unit name. This Run Option is also used to assign unit names when the Catalog Access Run Option is set to CAC=NO,NOVTOC or when the unit name cannot be determined, such as a DEFINE IDCAMS control statement.

Use this run Option if the default unit names used by JOB/SCAN are not defined in the system (see [Parameters](#)).

These names must be coded in the Control Table AA ([Control Tables](#)) if you are running with the Catalog Access Run Option set to NO, NOVTOC, or if IBM module IEFEB4UV is not available on the system.

### Format



### Example

```
DEFAULT UNIT NAMES =DASD=3380 ,MASSTOR=3330V ,CARTRIDGE=CART
```

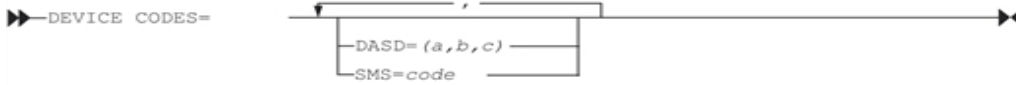
### Parameters

Parameter	Default	Function
DASD	3380	Specifies unit name to be assigned to all datasets residing on direct-access storage devices (except mass- storage devices).
TAPE	TAPE	Specifies unit name to be assigned to all datasets residing on tape reels.
MASSTOR	3330V	Specifies unit name to be assigned to all datasets residing on mass storage devices.
CARTRIDGE	CART	Specifies unit name to be assigned to all datasets residing on cartridge tapes.

## Device Codes (DVC)

This Run Option adds to or overrides alphabetic device codes in JOB/SCAN. Specify up to three additional one-letter alphabetic device codes, each to be treated as direct-access, report, or tape devices.

### Format



## Example

```
DEVICE CODES =DASD= (A, B) , SMS=D
```

## Parameters

Parameter	Default	Function
DASD	D only	Specifies up to three additional one-letter device codes (in addition to D) to be treated as direct-access devices. UNITS added with this parameter are included in the Space Analysis. See <a href="#">Additional Device Codes</a> .
SMS	This parameter defaults to the device code derived from the system or from the Control Table AA entry as noted in Function.	Specifies a device code that is assigned to all system-managed storage (SMS) datasets. The device code specified in the Control Table AA entry for the UNIT name specified in the JCL or derived from the system by JOB/SCAN, is overridden by this device code and will be treated as a DASD device. If this parameter is omitted or null, the device code is assigned from Control Table AA. If there is no Control Table AA entry for the UNIT, a device code of D is assigned.

## Additional Device Codes

Do not code these reserved device codes in the DASD parameter:

C	Cards	M	Mass storage	V	VIOFX
D	Direct access	R	Report	X	Dummy
G	Graphics	T	Tape	Z	User defined code
K	Cartridge tape	U	Unknown device		

## DMS Volume Serial (DMSV)

This Run Option defines the volume serial in the MVS catalog in which datasets are archived. Use DMS Volume Serial with the Program Product Interface Run Option set to DMS:

```
PROGRAM PRODUCT INTERFACE=DMS
...
DMS VOLUME SERIAL=ASG001
```

The volume serial number of the DMS Volume Serial Run Option must equal the volume serial coded in the RECATVOLVvvvvvv parameter of DMS SYSPARMS.

### Format

▶▶—DMS VOLUME SERIAL= *nnnnnn* —▶▶

### Example

DMS VOLUME SERIAL=AMS000

### Parameters

Parameter	Default	Function
<i>nnnnnn</i>	ARCIVE	Specifies the volume serial number in the MVS catalog to which a dataset that has been archived is now cataloged.

## Exec Parm String (EPS)

This Run Option directs JOB/SCAN Change to string together a multi-line parm string. This allows a mask to be applied to the entire parameter string.



When the parm string is combined, the change is permanent.

### Format

▶▶—EXEC PARM STRING=  NO  YES —▶▶

### Example

EPS=YES

### Parameters

Parameter	Default	Function
YES/NO	NO	YES causes the multi-line parameter from an exec statement to be combined into one string, thereby allowing a mask to be applied to the entire parameter string. NO prevents this action.

## GDG Processing (GDGP)

This Run Option controls the external tracking of Generation Datasets to conform to user site conventions for GDG control. This would be used by sites that have modified their system such that the GDG Base Record Association is updated at the end of each step (that is, the +1 refers to the same dataset only within the step). Most sites will be able to use the default.

### Format



### Example

```
GDG PROCESSING=ENDOFSTEP
```

### Parameters

Parameter	Default	Function
ENDOFJOB/ ENDOFSTEP	ENDOFJOB	ENDOFJOB specifies that +1 datasets should be internally cataloged at the end of the job. ENDOFJOB may be abbreviated to EOJ. ENDOFSTEP specifies that +1 datasets should be internally cataloged at the end of the step. ENDOFSTEP may be abbreviated to EOS. Before specifying ENDOFSTEP, verify with the system programmer at the installation that the operating system catalogs GDGs at the end of a step.

## Group ID (GRP)

This Run Option specifies a value for any &GRP symbolics coded in a DSN= parameter in File Allocation statements. (See [File Allocations Group](#) for information on coding File Allocation statements.)

### Format



### Example

This example shows how the GRP Run Option and the File Allocations DSN work together. These two statements when combined:

```
FILE ALLOCATIONS
```

```

          JLSCRN      =DSN=&GRP..JLSCRN, SHR
RUN OPTIONS
          GROUPID=USRGL11

```

have the same effect as this statement:

```

FILE ALLOCATIONS
  JLSCRN      =DSN=USRGL11.JLSCRN, SHR

```

## Parameters

Parameter	Default
Groupid value	The RACF groupid.

## Headings (HDG)

This Run Option specifies report heading formats. This Run Option affects all reports.

### Format



### Example

```
HEADINGS=TERSE
```

### Parameters

Parameter	Default	Function
VERBOSE/TERSE	VERBOSE	<p>VERBOSE specifies column heading descriptions are printed on every page of each report.</p> <p>TERSE specifies column heading descriptions are printed on only the banner page of each report.</p>

## HSM Volume Serial (HSMV)

This Run Option defines the volume serial (VOLSER), found in the MVS catalog, indicating archived datasets. When JOB/SCAN receives this value in response to its request for a dataset VOLSER, it does not use the value as an actual VOLSER; it instead understands that the data set is archived. Use this Run Option in conjunction with the PROGRAM PRODUCT INTERFACE Run Option, HSM parameter.

## Format

```
▶—HSM VOLUME SERIAL=nnnnnn —▶
```

```
HSM VOLUME SERIAL=nnnnnn
```

## Example

```
HSM VOLUME SERIAL=HSM000
```

## Parameters

Parameter	Default	Function
nnnnnn	MIGRAT	This alphanumeric string specifies the VOLSER used by HSM to indicate archived datasets when the string is found in the MVS catalog. MIGRAT is HSM's default value indicating archived datasets.

## ISPF Specifications (ISS)

This Run Option determines whether functions requiring APF-authorization should be performed. It also specifies the type of format (TSO or non-TSO) for entering and displaying dataset names on JOB/SCAN ISPF panels. This option is only valid for Foreground processing.

## Format

```
▶—ISSP SPECIFICATIONS= [NOAUTHFUNCTIONS | AUTHFUNCTIONS | DSNFORMAT=TSO | DSNFORMAT=NONTSO] —▶
```

## Example

```
ISSP SPECIFICATIONS=NOAUTHFUNCTIONS, DSNFORMAT=NONTSO
```

## Parameters

Parameter	Default	Function
AUTHFUNCTIONS/ NOAUTHFUNCTIONS	AUTHFUNCTIONS	Specifies whether or not to allow JOB/SCAN to invoke optional APF-authorized functions under ISPF.



Parameter	Default	Function
DSNFORMAT=TSO/ DSNFORMAT=NONTSO	DSNFORMAT=TSO	The DSNFORMAT=TSO parameter specifies that dataset names can be entered on panels in standard TSO format. DSNFORMAT=NONTSO requires that dataset names be entered as fully qualified dataset names <i>not</i> enclosed in apostrophes.

### Additional Guidelines

When running under ISPF, APF-authorized functions might decrease performance. Bypass them when under ISPF by coding the NOAUTHFUNCTIONS parameter on this Run Option statement.

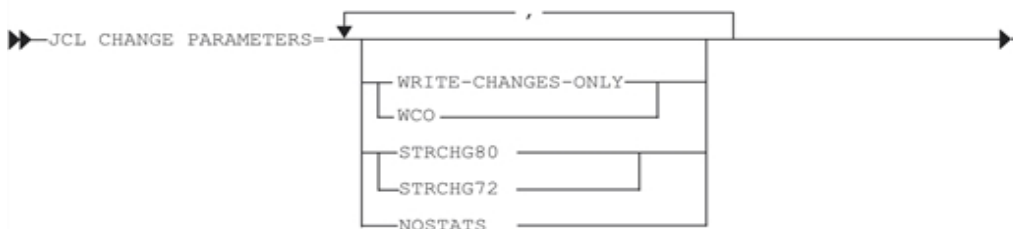
If TSO format is in effect, the first node of the dataset name should be omitted when it is the same as the TSO prefix. If the first node is not the same as the prefix, the fully qualified dataset name must be enclosed in apostrophes.

See [Running Jobs in an APF-Authorized Environment](#) for more information about running programs in APF-authorized mode.

## JCL Change Parameters (JCLCP)

This Run Option controls how JOB/SCAN JCL Change facility runs.

### Format



### Example

```

JCLCP=STRCH72
JCLCP=WCO, STRCH72

```

### Parameters

Parameter	Default	Function
WCO or WRITE-CHANGES-ONLY	JOB/SCAN writes all members seen by JCL Change.	Specify WCO to cause JOB/SCAN to write only those members it changed; omit WCO to cause JOB/SCAN to write all members.
NOSTATS		Prevent updating ISPF statistics when writing changed

Parameter	Default	Function
STRCHG72 STRCHG80	STRCHG80	<p>or reformatted JCL. Default is to write statistics.</p> <p>STRCHG 72 restricts the STRCHG command to columns 1-72; STRCHG80 allows the STRCHG command to work in columns 1-80.</p>

## JES2 Command Character (J2C)

This Run Option specifies the character the installation has selected as the JES2 Command Character. The IBM standard character is the dollar sign (\$). If the site has chosen a different character, consult with the JES2 systems programmer for the replacement character, and specify it with this statement.

### Format

```
► JES2 COMMAND CHARACTER=x ◀
```

### Example

```
JES2 COMMAND CHARACTER=&
```

### Parameters

Parameter	Default
x	\$

## JLJPA PROCLIB Type (JPAPT)

The JLJPA PROCLIB Type Run Option tells JOB/SCAN the library type (PDS, PANVALET, or LIBRARIAN) of the dataset in the JLJPA File Allocation statement. (See *Generating Default JLJPA Specifications* for more information about the JLJPA run option.)

### Format

```
► JLJPA PROCLIB TYPE=
┌ PDS
├ PANVALET | PAN
├ LIBRARIAN | LIB
└ LIBRARIAN-3 | LIB-3 ◀
```

## Example

```
JLJPA PROCLIB TYPE =PANVALET
```

## Parameters

Parameter	Default
PDS/ PANVALET/ LIBRARIAN/ LIBRARIAN-3	PDS

## JLJPA PROCLIB TYPE and SUBSYS File Allocation Parameter

When the SUBSYS File Allocation parameter is used, the JLJPA PROCLIB TYPE parameter should be set to PDS (the default setting). Code either:

```
RUN OPTIONS
FILE ALLOCATIONS
        JLJPA PROCLIB TYPE=LIB3
        JLJPA=SHR, DSN=HLQ.LIBR.MASTER
```

Or

```
RUN OPTIONS
FILE ALLOCATIONS
        JLJPA PROCLIB TYPE=PDS <== default value
        JLJPA=SHR, DSN=HLQ.LIBR.MASTER, SUBSYS=LAM
```

## MVS Restriction

MVS does not allow dynamic concatenation of subsystem datasets. JLJPA must be statically pre-allocated (through JCL) when datasets are concatenated and at least one of the datasets are subsystem datasets. Make sure to code the SUBSYS parameter for all datasets, even the PDS datasets. For example:

```
//JLJPA DD DSN=ASGUSR.LIBR.MASTER, SUBSYS=LAM <== LIBR
// DD DSN=SYS1.PROCLIB, DISP=SHR, SUBSYS=LAM <== PDS
// DD DSN=ASGUSR.LIBR.NOPSWD, SUBSYS=LAM <== LIBR
```



- When datasets are pre-allocated with the SUBSYS JCL parameter, do not code a JLJPA PROCLIB TYPE Run Option statement.
- If a mixed subsystem type is used for JLJPA, the allocations of the JLJPA must be in your JCL and not included in JSOPT02.

## JLJPC PROCLIB Type (JPCPT)

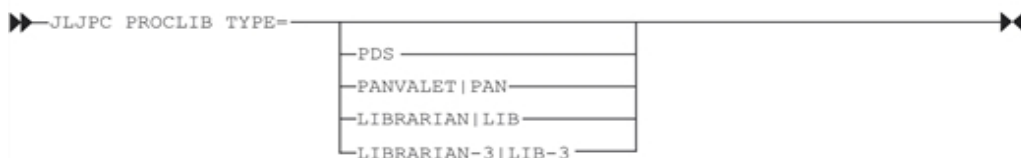
This Run Option tells JOB/SCAN the library type (PDS, PANVALET, or LIBRARIAN) of the dataset(s) in the JLJPC File Allocation statement.

The JLJPC procedure library name and JLJPC PROCLIB type are specified in the First PROCLIB field of the JCL Validation panel when processing under ISPF. Values entered using the panel override any parameters coded in this Run Option.



When executing JOB/SCAN in batch, this Run Option should be coded in an SJJSPARM library member. You will also need to add a JLJPC ddname to the batch JCL (JSMVS or JS6PROC), or enter the ddname in a File Allocation statement.

## Format



## Example

```
JLJPC PROCLIB TYPE=PANVALET
```

## Parameters

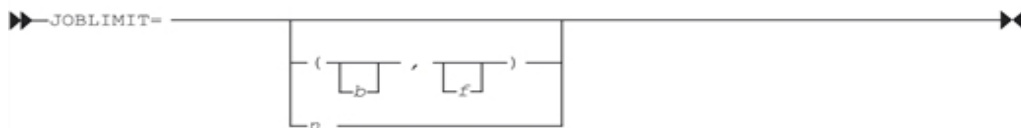
Parameter	Default
PDS/ PANVALET/ LIBRARIAN/ LIBRARIAN-3	PDS

## Job Limit

This Run Option limits the maximum number of jobs processed in a JOB/SCAN run. This Run Option prevents inadvertent or unauthorized use of the SELECT=ALL parameter.

When running JOB/SCAN from the ISPF panels, this option can be entered on the Processing Options panel rather than by coding this Run Option. JOB/SCAN compares the value on the panel to the corresponding Run Option from the SJJSPARM library, and uses the smaller of the two.

## Format



## Parameters

Parameter	Default	Function
<i>b</i>		The limit for batch processing mode; default is 99998. A value of zero allows an unlimited number of jobs.
<i>f</i>		The limit for foreground processing mode; default is 99998. A value of zero allows an unlimited number of jobs.
<i>n</i>	<i>n</i>	The limit for both batch and foreground processing modes; default is 99998. A value of zero allows an unlimited number of jobs.

## Example

This example has a batch limit of 50 and the foreground limit is 20:

```
JOB LIMIT=(50,20)
```

This example has an unlimited batch limit; and 75 as the foreground limit:

```
JOB LIMIT=(,75)
```

In this example both batch and foreground limits are 10:

```
JOB LIMIT=10
```

## JOB/SCAN PROC (JSP)

This Run Option specifies the name of the PROC used for executing JOB/SCAN in batch mode from the ISPF panels. The default PROC delivered on the installation tape is JS6PROC. Values in this PROC are modified at installation, and the PROC is stored in a system or test PROCLIB. If a different member name is used, code this statement with the new name.

## Format

```
▶▶JOB/SCAN PROC=xxxxxxxx ▶▶
```

## Example

```
JSP=WXYPROC
```

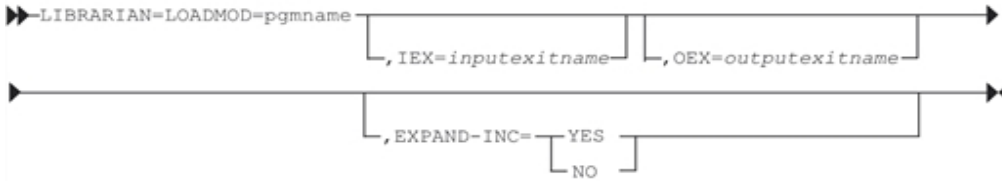
## Parameters

Parameter	Default	Function
xxxxxxxx	JS6PROC	The name of the batch PROC.

## LIBRARIAN (LIBR)

The LIBRARIAN Run Option specifies the CA-LIBRARIAN options needed by JCL Change or Reformatter. The options are the CA-LIBRARIAN load module name to invoke when writing output to a CA-LIBRARIAN master file, and any input or output exits for the CA-LIBRARIAN parameter field. It also specifies whether or not the -INC statements in the input JCL are expanded.

### Format



### Example

```
LIBRARIAN =LOADMOD=LIBRARIAN
```

### Parameters

Parameter	Default	Function
LOADMOD= <i>pgmname</i>	AFOLIBR	Specifies CA-LIBRARIAN program name assigned to CA-LIBRARIAN during the installation of that product.
IEX	no default	Specifies an optional CA-LIBRARIAN <i>inputexitname</i> .
OEX	no default	Specifies an optional CA-LIBRARIAN <i>outputexitname</i> .
EXPAND-INC	YES	<p>YES Code this parameter to cause FAIRREC to expand -INC statements and pass the expanded JCL to JCL Change and Reformatter. The -INC statements will not be passed.</p> <p>NO Code this parameter to cause FAIRREC to pass the -INC statements to JCL Change and Reformatter without expanding them.</p>



The EXPAND-INC parameter is ignored for JOB/SCAN validation. It affects only JCL Change and Reformatter when the input is from a CA-LIBRARIAN master file.



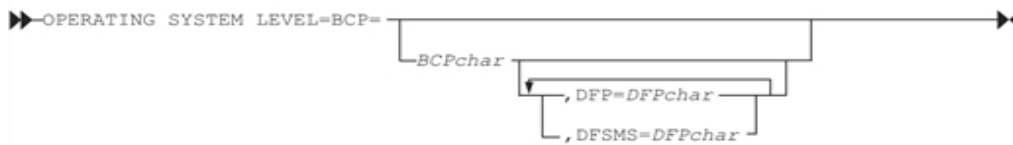
## Parameters

Parameter	Default
DISK/TAPE	DISK

## Operating System Level (OSL)

The Operating System Level Run Option allows validation of JCL for an operating system level different from the operating system JOB/SCAN is running on. It overrides the Base Control Program level and the Data Facility Product level currently running on the system.

### Format



### Example

```
OPERATING SYSTEM LEVEL=BCP=4,DFSMS=J
```



## Parameters

Parameter	Default	Description and values
		Base Control Program level for which the JCL is to be validated:
		U=z/OS V1R1
		V=z/OS V1R2
		W=z/OS V1R3
		X=z/OS V1R4
		Y=z/OS V1R5
		Z=z/OS V1R6
		0=z/OS V1R7
		1=z/OS V1R8
		2=z/OS V1R9
		through z/OS
		V1R11
		3=z/OS V1R12
		4=z/OS V1R13
		5=z/OS V2R1
		6=z/OS V2R2
		7=Z/OS V2R3
		L=OS/390 1.2
		M=OS/390 1.3
		N=OS/390 2.4
		O=OS/390 2.5
		P=OS/390 2.6
		Q=OS/390 2.7
		R=OS/390 2.8
		S=OS/390 2.9
		T=OS/390 2.10
		D=MVS SP 3.2 and below
		E=MVS SP 4.1
		F=MVS SP 4.2
		G=MVS SP 4.3
		H=MVS SP 5.1
		I=MVS SP 5.2
		J=MVS SP 5.2.2
		K=OS/390 1.1
<i>BCPchar</i>	Current Base Control Program release level	
		Data Facility Product level for which the JCL is to be validated:
		C=DFP 2.3 and below
		D=DFP 3.1
<i>DFPchar</i>	Current Data Facility Product release level	
		Data Facility Product level for which the JCL is to be validated:
		E=DFSMS 1.1.0
		F=DFSMS 1.2.0
		H=DFSMS 1.4.0
		G=DFSMS 1.3.0
		I=DFSMS 1.5.0
		J=DFSMS R10 and higher
<i>DFSMSchar</i>	Current Data Facility Storage management subsystem release level	



There is no need to code DFP/DFSMS, if the release installed at the user's installation is DFSMS V2R10, or greater. Starting with DFSMS V2R10, the DFSMS product is shipped by IBM with OS/390 and z/OS390 without a separate DFSMS release number.

## Output Specifications (OSS)

This Run Option controls various aspects of the format of JOB/SCAN printed output.

### Format



### Example

```
OUTPUT SPECIFICATIONS=NOFOOT,NOBANNERPAGE
```

### Parameters

Parameter	Default	Function
BANNERPAGE/ NOBANNERPAGE	BANNERPAGE	Specifies whether or not a banner page is printed.
FOOT/NOFOOT	FOOT	Specifies whether a footer line appears as the last line on each printed page.
NOEURODATE/ EURODATE/ INTERNATIONALDATE	NOEURODATE	Causes the date to appear in MM/DD/YY format. If EURODATE is specified, the format is YY/MM/DD. If INTERNATIONALDATE is specified, the format is YYYY-MM-DD

## Override Indicator (ORI)

This Run Option specifies the character to use to highlight the override JCL statements in the Structured JCL Listing.

JOB/SCAN automatically highlights override DD statements by replacing the usual slash characters (//) with dollar signs (\$\$) as the statements are printed in the Structured JCL Listing. When the override statement is printed in the PROC following the DD statement to which it applies, the // is replaced by \$/. The character specified in this Run Option prints in place of the \$ on the override statements in the Structured JCL Listing.

### Format

```
►►—OVERRIDE INDICATOR=x —————►►
```

### Example

```
OVERRIDE INDICATOR=&
```

### Parameters

Parameter	Default	Function
x	\$	Specifies the character to replace the dollar sign (\$) in the override statements in the Structured JCL Listing. Do not use delimiters equals (=), comma (,), or parentheses (()).

## Prefix (PFX)

This Run Option specifies a new value for any &PF symbolics coded in a DSN= parameter in File Allocation statements. (See [System Administrator Tasks](#) for information on coding File Allocation statements.) If the site uses batch mode to document jobs, and &PF is in the File Allocation statements, code the Prefix Run Option.

### Format

```
►►—PREFIX=prefix-value —————►►
```

### Example

```
FILE ALLOCATIONS
      JLSCRN=DSN=&PF..JLSCRN, SHR
RUN OPTIONS
      PFX =USRPYR1
```

This is result:

```
FILE ALLOCATIONS
      JLSCRN=DSN=USRPYR1.JLSCRN, SHR
```

## Parameters

Parameter	Default	Function
<i>prefix-value</i>	See Purpose at right.	Specifies the value (eight characters or less) to be substituted for &PF in the dataset name. In interactive mode, &PF is set to the value of the TSO PROFILE PREFIX.  In batch mode, any &PF symbolics coded in File Allocation statements will be replaced with a null value if the Prefix Run Option is not coded.

## Processing Specifications (PSS)

This Run Option controls the effect of CONDCODE parameters during JOB/SCAN execution. JOB/SCAN ISPF users can enter these values on the Processing Options panel instead of coding this Run Option. Values entered on the panel override corresponding parameters of this Run Option coded in the SJJSPARM library.

### Format



### Example

```
PROCESSING SPECIFICATIONS=CONDCODE
```

## Parameters

Parameter	Default	Function
NOCONDCODE/ CONDCODE	NOCONDCODE	Simulates condition code processing for each step in a job. By coding "CONDCODE", the COND= parameter of each step is evaluated and run-time validation is performed accordingly. If a COND= parameter is found to be true, the step is syntax-checked and appears in the Structured JCL listing, but run-time validation does not occur. Any files used or deleted in the affected steps are not tracked.

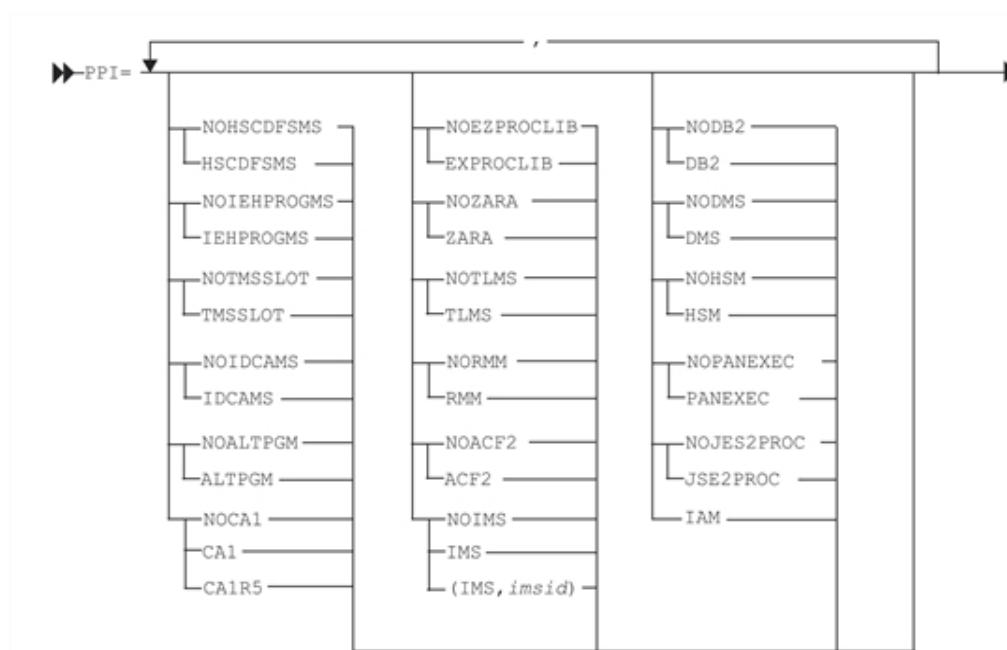
### Additional Guidelines

- Each step is assigned a return code of 0 unless a Control Table AR entry is specified.
- CONDCODE excludes all program steps identified by Control Table AU entries. Steps that are excluded are ignored entirely.
- When a Control Table AR entry match is found, an advisory message is printed that states the return code determined for this step. If no Control Table AR entry is found, no message is issued.
- When a COND= parameter match for a given step is found, an advisory message is issued that notes the match and the parameter value.
- Condition code validation is done for COND= parameters on the job statement, EXEC PROC statement, and EXEC PGM statements, as well as for // IF statement conditions.

## Program Product Interface (PPI)

This Run Option specifies the products with which JOB/SCAN should interface.

### Format



### Example

PROGRAM PRODUCT INTERFACE=CA1, TMSSLOT

## Parameters

Parameter	Default	Function
NOACF2/ACF2	NOACF2	Used to change the wording in message DSS4757 as follows: NOACF2 ACF2 READ READ UPDATE WRITE ALTER ALLOCATE
NOALTPGM/ ALTPGM	NOALTPGM	Useful for jobs executing products such as SAS FOCUS RAMIS, etc. If ALTPGM is coded here and there is a corresponding entry in Control Table AD, JOB/SCAN uses the member specified in the DD statement as the program name.
NOCA1/CA1/ CA1R5	NOCA1	Uses CA1 TMS catalog to obtain tape dataset information. Specify CA1R5 if TMS catalog is at CA1 Release 5.0 or greater. If CA1 or CA1R5 is specified, Control Table AM is searched for out-of-area codes.
NODB2/DB2	NODB2	If DB2 is specified, turns on the DB2 interface.  If you are interfacing with batch DB2 commands, you may need to update Control Table AB (see <a href="#">Control Table AB: Control Card DD Name</a> ).
NODMS/DMS	NODMS	If using the DMS/OS DASD management system, DMS should be specified. If DMS is specified, a warning message is issued for datasets with a volume serial of ARCIVE, indicating that the datasets are archived. If DMS is not specified, unexpected results may occur. This may cause a delay when the job is submitted to the system. Related Run Options are DMS Volume Serial (DMSV) and Migrate Device Type (MDTP).
NOEZPROCLIB/ EZPROCLIB	NOEZPROCLIB	If EZPROCLIB is specified, PROCs are retrieved from libraries specified on PROCLIB, DYNAPROC, or JOBPROC DD statements. The JLJPB ddname specifies the libraries. (See <a href="#">Searching for a PROC During Validation</a> .)
NOHSCDFSMS/ HSCDFSMS	NOHSCDFSMS	Specify HSCDFSMS if your installation uses StorageTek's HSC/DFSMS Interface to control allocation for datasets to be created on cartridge tape.
NOHSM/HSM	NOHSM	If using the HSM DASD management system, HSM should be

Parameter	Default	Function
		specified. If HSM is specified, a warning message is issued for datasets with a volume serial of MIGRAT, indicating the datasets are migrated. If HSM is not specified, you may get unexpected results. This may cause a delay when the job is submitted to the system. Related Run Options are HSM Volume Serial (HSMV) and Migrate Device Type (MDTP).
IAM	n/a	Enables JOB/SCAN support for IAM (Innovation Access Method). The support is disabled by default.
NOIDCAMS/ IDCAMS	NOIDCAMS	If IDCAMS is specified, tracks of datasets created or deleted by an IDCAMS step in the same way JCL references are tracked. JOB/SCAN ISPF users can enter this parameter on the Processing Options panel rather than by coding this Run Option parameter. The value entered on the panel overrides any corresponding parameter coded in the SJJSPARM library. (See also descriptions of Control Tables AB and AQ in <a href="#">Control Tables</a> ).
NOIEHPROGM/ IEHPROGM	NOIEHPROGM	If IEHPROGM is specified, turns on the IEHPROGM interface. (See the description of Control Table AD in the <a href="#">Control Tables</a> ).
NOIMS/IMS/  (IMS,imsid)	NOIMS	If IMS is specified, it turns on the IMS interface. (See also descriptions of Control Tables AD and AP in the <a href="#">Control Tables</a> ).  Turns on IMS interface and specifies a one- to four- character name for the IMS Subsystem Name. The order of priority of this ID is JCL (IMSID in the DFSRRC00 PARM field), then this PPI IMS parameter. There is no default for <i>imsid</i> . Example: (IMS,X16H)
NOJES2PROC/ JES2PROC	NOJES2PROC	If JES2PROC is specified, JOB/SCAN uses the JLJPA allocations from JSOPTJ2 instead of JSOPT02.
NOPANEXEC/ PANEXEC	NOPANEXEC	If PANEXEC is specified, allocates any PECNTL DD statements found in the job being processed. In this way, proper control information is made available to the PANEXEC module for verifying program names in load libraries under control of PANEXEC.
NORMM/ RMM	NORMM	If RMM is specified, the interface to RMM (IBM's Removable Media Manager) is enabled to obtain volume and dataset information. RMM must be installed and running to use this option. If RMM is specified, Control Table AM is searched (if provided) for Stor or Loan locations considered on-site.

Parameter	Default	Function
NOTLMS/ TLMS	NOTLMS	JOB/SCAN can access the CA-DYNAM/TLMS tape management system Volume Master File (VMF) to verify that volume and dataset information coded in the JCL or obtained from the system catalog matches what is in the VMF. After the TLMS interface is installed, JOB/SCAN automatically invokes it as necessary.
NOTMSSLOT/ TMSSLOT	NOTMSSLOT	If TMSSLOT is specified as well as CA1 or CA1R5, enables slot number verification.
NOZARA/ZARA/ (ZARA, <i>subsystem name</i> )	NOZARA	If ZARA is specified, the interface to ZARA is enabled to obtain tape dataset information. JOB/SCAN uses ZARA as the name of the ZARA subsystem. If your installation uses another name, then code it as the second subparameter.

## Report Heading

This Run Option specifies a user-defined title for each page of output. When running foreground, the report heading on the JOB/SCAN - JCL Validation panel is used rather than the heading coded in this Run Option.

### Format

►—REPORT HEADING=*string* —◄

### Example

```
REPORT HEADING='MONTHLY PAYROLL'
```

### Parameters

Parameter	Default	Description
'string'	No default	1- to 32-character phrase enclosed between single quotation marks. Do not embed any single quotation marks in the phrase.

## Resolve System Symbols (RSS)

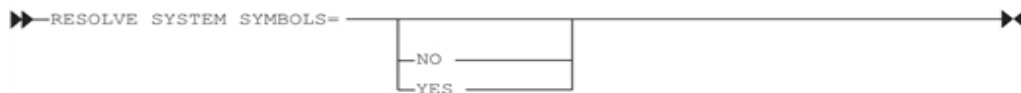
The Resolve System Symbols Run Option identifies that the input JCL stream is to be processed as a started task or TSO Logon procedure. JOB/SCAN resolves the system variables such as &SYSNAME and &SYSCONE.





If running on an MVS system prior to MVS/SP 5.2.0, this Run Option is ignored. It is also ignored if the system symbol substitution service routine cannot be loaded.

### Format



### Example

```
RESOLVE SYSTEM SYMBOLS=YES
```

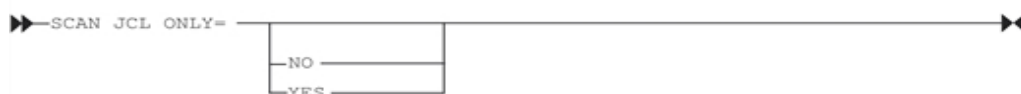
### Parameters

Parameter	Default	Function
YES/NO	NO	<p>Determines whether system symbols found in the JCL are to be resolved.</p> <p>Specify YES to cause input JCL to be processed as a started task or TSO Logon procedure. System symbols found in the JCL will be resolved.</p> <p>Specify NO to ignore system symbols.</p>

## Scan JCL Only (SJO)

This Run Option determines whether or not JOB/SCAN processes a member, if a slash (/) is not in column 1.

### Format



### Example

```
SCAN JCL ONLY=YES
```

### Parameter

Parameter	Default	Function
YES/NO	NO	<p>If SCAN JCL ONLY=YES is specified, JOB/SCAN processes only those members that have a slash (/) in column 1. If SCAN JCL ONLY=NO is specified, JOB/SCAN</p>





SECURITY EXIT=SAF2, DSNAME, USER1, GROUP1

**Parameters**

Parameter	Default	Function
xxxxxxxx	No default	The name of the security exit program interfacing JOB/SCAN with a security authorization system. This program must reside in the LINKLST library or, JOBLIB/STEPLIB library, or the load library allocated by ddnames JLLD or SJJSLMOD.
NODSNAME/ DSNAME	NODSNAME	The DSNAME parameter instructs the security exit program to check each dataset name in the input JCL for possible access violation.  <div style="border: 1px solid purple; padding: 5px; display: inline-block;"> <span style="color: purple;">★</span> DSNAME is not supported by the ACF2EXIT or RACFEXIT shipped in SJJSSAMP.                 </div>
userid	No default	Specifies the USERID that will ultimately submit the job for execution (TWS user, CA7 user, etc.) and is used only for the DATA SET NAME validation function of the JOB/SCAN security interface. This field will not override a USER parameter in the JOB statement of the job being scanned.
groupid	No default	Specifies the GROUPID that will ultimately submit the job for execution (TWS user, CA7 user, etc.) and is used only for the DATA SET NAME validation function of the JOB/SCAN security interface. This field will not override a GROUP parameter in the JOB statement of the field being scanned.

**SEMRESET**

This Run Option determines whether the Suppress Error Messages value in Foreground defaults to NO for each foreground JOB/SCAN session. (See the [Suppress Error Messages \(SEM\)](#)) This Run Option affects the foreground Processing Options panel.

**Format**



**Example**

SEMRESET =YES

## Parameters

Parameter	Default	Function
YES/NO	YES	<i>YES</i> causes the Suppress Error Messages value in Foreground to default to <i>NO</i> . If <i>NO</i> is specified, causes the Suppress Error Messages specified in Foreground Processing Options panel to be saved for subsequent ISPF sessions and not reset to <i>NO</i> for each session.

## Separator (SEP)

This Run Option specifies the character pair used to separate the heading of the reports.

### Format



### Example

```
SEPARATOR = (++, SPACENONE)
```

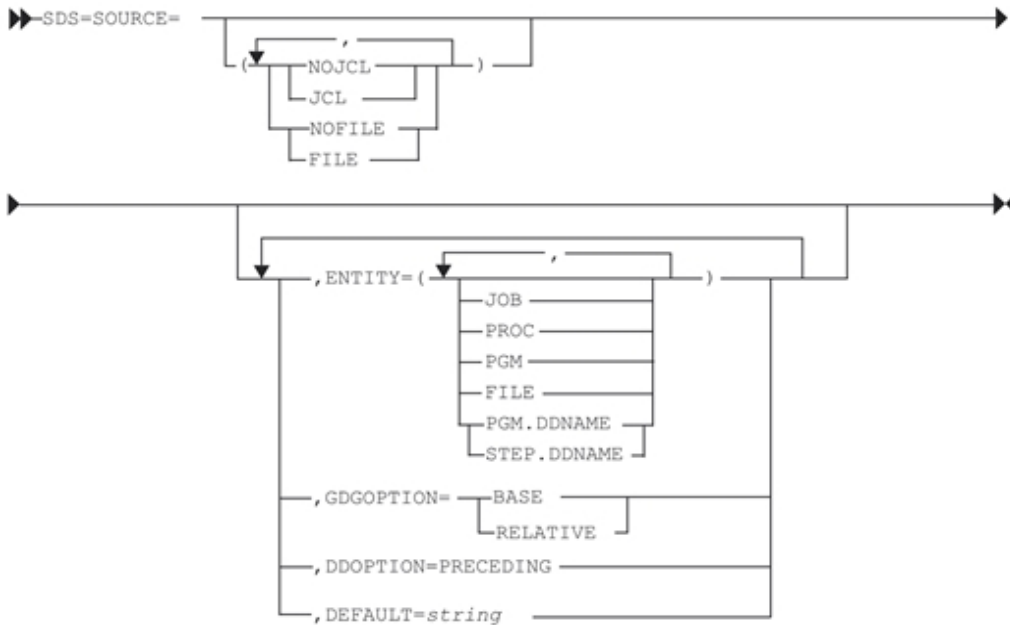
## Parameters

Parameter	Default	Function
xx	--	Specifies the two-character set to be used as the separator. Any alphanumeric characters can be used.
SPACEBEFORE/ SPACEAFTER/ SPACENONE	SPACEBEFORE	Inserts a blank line before or after each separator line. SPACENONE suppresses blank lines between blocks.

## Short Descriptions (SDS)

This Run Option associates a descriptive phrase of up to 32 characters with any JOB, PROC, PGM, FILE, PGM.DDNAME, or STEP.DDNAME. ASG-DOCU/TEXT, ASG-ONLINE XREF, and ASG-INFO/X customers who have defined short descriptions in an OLX/SDF VSAM file through the Short Descriptions Facility, can include these descriptions in JOB/SCAN reports. (See the DOCU/TEXT, ONLINE XREF, or INFO/X product documentation for complete information about the Short Descriptions Facility.)

**Format**



**Example**

```

SHORT DESCRIPTIONS=SOURCE=JCL
SHORT DESCRIPTIONS=SOURCE=(JCL, FILE)
    
```

**Parameters**

Parameter	Default	Function
SOURCE	JCL,NOFILE	<p>Specifies source of the Short Description text. If both sources are coded, source given first is searched first and used if a Short Description is found.</p> <ul style="list-style-type: none"> <li>JCL/NOJCL tells JOB/SCAN whether to use any comments in the JCL statements for Short Descriptions.</li> <li>FILE/NOFILE is applicable only if have defined Short Descriptions in an OLX/SDF VSAM file.</li> </ul> <p>Parentheses are required only if both sources are coded.</p>
ENTITY	JOB,PROC, PGM, FILE, PGM.DDNAME	<p>Specifies levels for which Short Descriptions are to be printed:</p> <ul style="list-style-type: none"> <li>JOB includes job-level Short Descriptions in the reports.</li> <li>PROC includes procedure-level Short Descriptions in the reports.</li> <li>PGM includes program-level Short Descriptions in the</li> </ul>

Parameter	Default	Function
		<p>reports.</p> <ul style="list-style-type: none"> <li>• FILE includes file-level Short Descriptions in the reports.</li> <li>• PGM.DDNAME includes report names based on PGM.DDNAMEs in the JCL. The PGM entity must also be specified.</li> <li>• STEP.DDNAME includes report names based on STEP.DDNAMEs in the JCL. The PGM entity must also be specified.</li> </ul>
GDGOPTION= BASE RELATIVE	RELATIVE	<p>Specifies how the Short Description Prompter Panel displays a GDG dataset. With BASE coded (abbreviation B), the panel will display the GDG base name <i>without</i> the relative generation, i.e. (+1), number.</p> <p>With RELATIVE coded (abbreviation REL or R), the panel will display the GDG base name <i>with</i> the relative generation, i.e. (+1), number.</p>
DDOPTION= PRECEDING	No default	<p>Specifies use of the first 32 non blank characters of a comment (//*) statement immediately preceding a DD statement as the Short Description for that DD. The FILE entity must be specified.</p> <p>If no comment (//*) statement precedes the DD statement, any comment on the DD statement is used as the Short Description.</p> <p>If not specified, this option is of no effect.</p>
DEFAULT= 'string'	No default	<p>Specifies a default Short Description to be used if none is found (in the JCL or FILE) for the specified entity. This Short Description can be up to 32 characters long and must be enclosed in single quotations.</p> <p>If not specified, this option is of no effect.</p>

### Entering I/O Codes

The I/O Code field allows I/O values to be overridden for file-level JCL items defined elsewhere, such as Control Table AC or as determined from the disposition in the JCL.

When JOB/SCAN processes a DD statement, the comment field is checked for one of these two character sequence codes: +I, +O, +M, +R, or +U. If a sequence code is not specified, the ddname is checked for a match in Control Table AC (see [User Information - JOB/SCAN](#)). If a match is found, the indicator from the Control Table AC entry is used. If neither condition is met, the DISP parameter in the JCL determines file treatment.

### Valid I/O Codes

This table identifies valid I/O codes:

Code	Indicates the File Should Be Treated as
I (Input)	An INPUT file
O (Output)	An OUTPUT file
M (Mod)	A MOD file (will appear on both the INPUT and the OUTPUT side of the Flowcharts report)
R (Reference)	A REFERENCE file (will appear as a one-line entry at the start of the job step)
U (Update)	An UPDATED file (will appear on both the INPUT and the OUTPUT side of the Flowcharts report)

### Entering Device Codes

The Device Code field allows you to override the type of device for a particular file as determined from the system or as specified in Control Tables AA or AG. (Refer to [Control Tables](#) for information on Control Tables.)

Valid device codes are:

Code	Meaning
(space)	Default device
C	Card
R	Report (Printer)
V	VIO (Virtual I/O)
D	Disk
G	Graphics (Display)
M	Mass storage
T	Tape (Reel)
K	Cartridge
X	Dummy
xyz	Any of three possible user-defined device types that may be established with the Device Codes Run Option (see <a href="#">System Administrator Tasks</a> ).



## Entering User Text Pointers

The Short Description facility can be used to store User Text pointers in the SDF/OLX file. JOB/SCAN uses the pointers to locate User Text for files and SYSOUTs and User Text stored under alternate member names.

### To insert a pointer

- Type the User Text member name in the Member Name field. (Refer to Adding User Text pointers to JCL in DOCU/TEXT User Information for more information about User Text pointers.)



When defining I/O Code, Device Codes, and/or User Text in your JCL on a DD statement, these values must be placed on the first line and before any comments that are coded upon the DD statement. For example:

```
//DD1 DD DSN=.GLPROD.SJJSIMOD, +RD (Pointer) Production Load Lib
//      DISP=SHR
```

**Or**

```
//DD1 DD DSN=DSSI.GLPROD.SJJSIMOD,
//      DISP=SHR          +RD (Pointer) Production Load Lib
```

### Add Short Description Entries panel

```
----- ADD SHORT DESCRIPTION ENTRIES ----- JOB/SCAN
COMMAND ===>

A - Add Record      S - Skip Record

Type = DATA SET                      Record Count:
Name = DSSI.WORK.SJJSIMOD                1 of 4

Enter Short Description    ===> EMPLOYEE DATA
Enter I/O Code            ===> I
Enter Device Code         ===> T
Enter User Text Member Name ===> MYTEXT
```

The Add Short Description Entries panel can be used to specify I/O code, device code, and User Text member name.

## SMS

This Run Option determines whether JOB/SCAN accesses the Automatic Class Selection (ACS) routines and the Construct Access Services (CAS) of the MVS DFSMS subsystem. The DFSMS subsystem provides information needed to validate JCL DD statements used to allocate new datasets. The ACS routines provide selected constructs (DATA CLAS, STORCLAS, and MGMTCLAS names) just as MVS allocation obtains them. CAS verifies constructs specified in JCL and selected by ACS routines and provides dataset attributes (such as SPACE and DCB information). The SMS Run Option is ignored if the SMS subsystem is not active during JOB/SCAN execution.



If SMS is active on the system and NOACS or NOCAS is specified, verification might be unpredictable when the input JCL contains SMS keywords or relies on SMS for implied required parameters.

**Format**



**Example**

SMS=NOACS, NOCAS

**Parameters**

Parameter	Default	Function
ACS/NOACS	ACS if SMS is active; NOACS if inactive	Permits or prohibits JOB/SCAN from invoking the ACS routines.
CAS/NOCAS	CAS if SMS is active; NOCAS if inactive	Permits or prohibits JOB/SCAN from invoking the CAS.

**Span Members (SMB)**

This Run Option specifies whether one job comprises of two or more different members. If SPAN MEMBERS=NO is specified, a new member is assumed to be a new job. If SPAN MEMBERS=YES is specified, JOB/SCAN continues scanning for another job card and processes everything as though it belongs to the previous job.

**Format**



**Example**

SPAN MEMBERS=YES

If the Input Selection statement is:

Input Selection

```
DSN=xxx.xxx
SELECT=MEMBER1, MEMBER2
```

and MEMBER1 is:

```
//JOB1 JOB
//STEP1 EXEC PGM=IEFBR14
//DD1 DD DSN=xxx.xxx, DISP=SHR
```

and MEMBER2 is:

```
//STEP1 EXEC PGM=IEFBR14
//DD1 DD DSN=xxx.xxx, DISP=SHR
```

specifying SPAN MEMBERS=YES results in JOB/SCAN processing MEMBER1 and MEMBER2 as one job; otherwise with SPAN MEMBERS=NO results in JOB/SCAN processing MEMBER1 and MEMBER2 as two separate jobs, with JOB/SCAN generating a JOB card for MEMBER2.

## Parameters

Parameter	Default	Function
NO/YES	NO	Indicates whether or not one job can be comprised of two different members.

## Standards Name (STN)

This Run Option statement is used when running JOB/SCAN Edit macros or in batch using the Standards Facility. This Run Option identifies the program name of the user-written standards program. A standards name entered on the JOB/SCAN - JCL Validation panel overrides any parameter coded in this statement.



A Standards name coded in an Access Control Run Option statement overrides the Standards Name Run Option and will appear on the JOB/SCAN - JCL Validation panel as a forced standards program.

### Format

```
STANDARDS NAME= programname [,parmvalue]
```

### Example

```
STN=PGM006A, 'DVTSCOL'
```

## Parameters

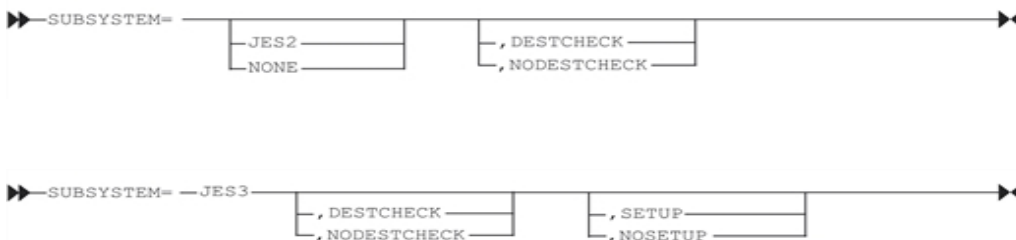
Parameter	Default	Function
<i>programname</i>	No default	Specifies name of the user-written standards program.
' <i>parmvalue</i> '	No default	Specifies an optional parameter to be passed to the program. Maximum size within the quotation marks is 32 characters.

## Subsystem (SSS)

This Run Option identifies the site-specific JES product for JOB/SCAN processing. If any of the JES statements or parameters are nonstandard, Control Table AS (described in [Control Tables](#)) may be used to list nonstandard JES statements and parameters to be excluded from error validation. The DESTCHECK option requires an APF-authorized environment.

JOB/SCAN ISPF users can enter these values on the Processing Options panel rather than by coding this Run Option. Values entered on the panel override corresponding parameters of this Run Option coded in the SJJSPARM library.

### Format



### Example

```
SUBSYSTEM=JES3
```


### Parameters

Parameter	Default	Function
JES2/JES3/NONE	JES2	Identifies whether JES2 or JES3 type statements are to be validated and passed to the standards program. NONE prevents the processing of JES2 or JES3 statements.
DESTCHECK/ NODESTCHECK	DESTCHECK	Permits or prohibits JES destination checking. (Requires APF-Authorized environment when running z/OS prior to V1R9. See <a href="#">Running the Product in an APF-authorized Environment.</a> )

Parameter	Default	Function
SETUP/NOSETUP	SETUP	JES3 only. Code NOSETUP if the JES3 STANDARDS initialization "SETUP=NONE" parameter is coded at your installation.

## Suppress Error Messages (SEM)

This Run Option suppresses, by severity level, error messages that would otherwise appear in the Structured JCL Listing. This Run Option also affects messages generated by the JOB/SCAN Standards Facility. JOB/SCAN ISPF users enter these values on the Processing Options panel rather than by coding this Run Option. Values entered on the panel override corresponding parameters of this Run Option coded in the SJJSPARM library (see [SEMRESET](#)).

 • validating from JOB/SCAN's ISPF panels, this option the Processing Options panel setting in the Suppress Error Message field overrides this run option.

• When validating from JOB/SCAN's Edit macros, the setting in JSOPTM (or local equivalent) overrides this run option.

### Format



### Example

SUPPRESS ERROR MESSAGES =04

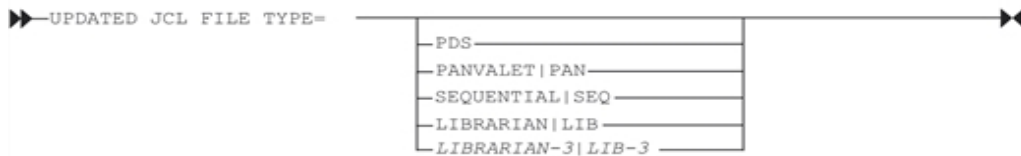
### Parameters

Parameter	Default	Function
NO/A/W/E/nn	NO	<p><b>NO</b> Prevents suppression of any messages.</p> <p><b>A</b> Advisory messages only.</p> <p><b>W</b> All advisory and warning messages.</p> <p><b>E</b> All advisory, warning, and error messages.</p> <p><b>nn</b> All messages with a severity level less than or equal to a number (nn=0 to 99).</p>

## Updated JCL File Type (UJFT)

This Run Option specifies the type of dataset to be written by Reformatter or JCL Change.

### Format



### Example

UJFT=PDS

### Parameters

Parameter	Default	Function
PDS/PANVALET/ SEQUENTIAL/ LIBRARIAN/ LIBRARIAN-3	PDS	<p>Specifies that Reformatter or JCL Change dataset (JLCHMO) is one of these datasets:</p> <ul style="list-style-type: none"> <li>• PDS (partitioned)</li> <li>• PANVALET</li> <li>• Sequential</li> <li>• LIBRARIAN</li> <li>• LIBRARIAN-3</li> </ul>

### Additional Guidelines

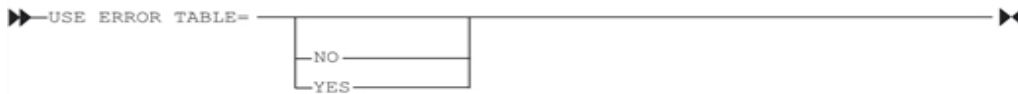
When running Reformatter or JCL Change in ISPF foreground or in panel-assisted batch, this Run Option is set from the panel.

## Use Error Table (UET)

This Run Option tells JOB/SCAN whether or not to search Control Table AW when generating the Structured JCL Listing. If this Run Option is set to YES, the Control Table AW entries either suppress error messages or change their severity level. If this Run Option is set to NO, Control Table AW is not used.

JOB/SCAN ISPF users enter these values on the Processing Options panel rather than coding this Run Option.

### Format



## Example

```
USE ERROR TABLE =YES
```

## Parameters

Parameter	Default	Function
NO/YES	NO	Tells JOB/SCAN to use the Control Table AW entries (YES) or to ignore Control Table AW (NO).

## Use PROC Exclude Table (UPCX)

This Run Option tells JOB/SCAN whether or not to search Control Table AN for PROCs to exclude from validation. If this Run Option is set to YES, all procedures listed in Control Table AN are excluded from the validation process.

## Format



## Example

```
USE PROC EXCLUDE TABLE =YES
```

## Parameters

Parameter	Default	Function
NO/YES	NO	YES excludes from validation all PROCs found in Control Table AN; NO ignores Control Table AN (NO).

## User ID (UID)

This Run Option specifies a value for any &UID symbolics coded in a DSN= parameter in File Allocation statements. If the site uses batch JCL mode to document jobs and &UID is in the File Allocation statements, code the User ID Run Option.

## Format

```
▶▶—USER ID=userid-value —————▶▶
```

## Example

These Run Options statements:

```
FILE ALLOCATIONS
  JLSCRN  =DSN=&UID..JLSCRN, SHR
RUN OPTIONS
  USER ID =USRGL11
```

are the same as this one:

```
FILE ALLOCATIONS
  JLSCRN  =DSN=USRGL11.JLSCRN, SHR
```

## Parameters

Parameter	Default	Function
<i>userid-value</i>	See Purpose at right.	Specifies the value given to any &UID symbolic in the dataset name. In interactive mode, &UID is set to the TSO USERID. In batch mode, &UID is set to the RACF ID, if available. If not available, it is set to the submitted job name less one character. For example: //ASGUSR JOB (ACCT)... would set &UID to ASGUSR.

## Write Selected JCL (WSJ)

This Run Option tells JOB/SCAN to write JCL being processed to ddname JLLAA so that the next step can submit this JCL if there are no errors. This Run Option applies to JOB/SCAN when executed in a batch environment.

### Format

```
▶▶—WRITE SELECTED JCL= —————▶▶
```

NO	_____
YES	_____

## Example

```
WRITE SELECTED JCL=YES
```



## Parameters

Parameter	Default	Function
NO/YES	NO	YES specifies that processed JCL is written to dataset JLLAA and subsequently submitted to MVS for execution; NO specifies otherwise.

---

# Control Tables

Control Tables specify how JOB/SCAN processes JCL. For example, Control Table AQ identifies specific DD names that reference control card libraries. Each table is uniquely identified by a two-character code. The Control Table values are usually set at installation time, but can be modified whenever necessary.

- Control Table-interaction with Run Options
- Control Table AA: Device Type, Device Group, and Unit Address
- Control Table AB: Control Card DD Name
- Control Table AC: Input/Output DD Names
- Control Table AD: Utility Programs
- Control Table AM: Tape Management Locations (Out-of-Area codes)
- Control Table AN: Procedures Excluded from Validation
- Control Table AP: PARM Programs
- Control Table AQ: Control Card Dataset Names
- Control Table AR: Return Code Simulation
- Control Table AS: Nonstandard JES Statements
- Control Table AT: Nonstandard OUTPUT Keywords
- Control Table AU: Job Steps to Ignore
- Control Table AW: Suppress JCL Error Messages
- Control Table AX: Nonstandard Parameters for DD Statements
- Control Table AY: Nonstandard Parameters for EXEC Statements
- Control Table AZ: Nonstandard Parameters for Job Statements
- Control Table BA: Reformatting DD Statements
- Control Table BB: Reformatting EXEC Statements
- Control Table BC: Additional Reformatter Options
- Control Table BD: Reformatting PROC Statements
- Control Table BE: Reformatting JOB Statements
- Control Table BF: Reformatting OUTPUT Statements
- Control Table BG: Reformatting SET Statements
- Control Table BH: Reformatting JCLLIB Statements
- Control Table BI: Reformatting INCLUDE Statements
- Control Table CB: Program Name Exit
- Control Table CC: VOLSER
- Control Table CD: DB2 Group Attachment Names

- Control Table CE: Scheduler Restart Program Names
- Control Table CF: Subsystem Dataset Disposition
- Control Table CG: Datasets That Are Not Control Card Datasets
- Control Table HF: HFS Path Name DD statements
- Control Table Ux: Standards Facility User Table

## Control Table-interaction with Run Options

Some Control Tables have an associated Run Option statement that must be coded in order to use the table. Other Run Option statements allow users to control the way in which a Control Table affects JOB/SCAN operation. These interaction are noted in the descriptions of Control Tables in the remainder of this topic.



Overriding SJJSPARM defaults does not apply to Control Tables with JOB/SCAN using the first table entry it encounters, and ignoring duplicates. To cancel previous Control Table entries, you might need to contact the JOB/SCAN system administrator to remove them from the SJJSPARM member in which they reside.

## Control Table AA: Device Type, Device Group, and Unit Address

Control Table AA specifies each device type, device group, or unit address that can appear in the UNIT= parameter of a DD statement. A device code is assigned to each unit name that identifies the type of device. For example, a unit name of DSS80 may be defined as a DASD device. If there is a UNIT= parameter in the JCL and JOB/SCAN cannot find the value defined to the system or in Control Table AA, the device is flagged as unknown.

### Associated Run Options

These Run Options are associated with Control Table AA:

<b>Default Unit Names (DUN)</b>	This Run Option assigns unit names to DASD, mass storage, reel tape, and cartridge tape devices for cataloged datasets in cases where JOB/SCAN runs on a system that does not have IBM module IEFEB4UV available. Control Table AA must contain the entries defined in this Run Option (or their default names).
<b>Device Codes (DVC)</b>	This Run Option allows up to three device codes to be specified for overriding or adding to existing devices codes. (See <a href="#">Conditions Requiring AA Entries</a> for additional information.)
<b>Catalog Access (CAC)</b>	This Run Option controls access to the system catalog and VTOCs (in the case of disk datasets). If this statement is set to NO, JOB/SCAN does not access the catalogs or VTOCs. (See <a href="#">Conditions Requiring AA Entries</a> for additional information.)

## Conditions Requiring AA Entries

Control Table AA entries are required if:

- The Catalog Access Run Option is set to NO,NOVTOC.
- The IBM Unit Verification Service routine IEFEB4UV is not available on the operating system.
- User-defined device codes are assigned to unit names in the Device Codes Run Option.
- A unit name is valid in the JCL but not defined to MVS using the Eligible Device Table (EDT).
- Customer site requires additional device types to be identified.

Include all valid device types, group names, or unit addresses. If other names in Control Table AA are not valid for the user site, remove them if desired. Standard Device Codes

This table shows device codes that may be entered in Control Table AA:

Code	Meaning	Remarks
C	Cards	None.
D	Direct access (disk)	Mandatory when Control Table AA is required.
G	Graphics (display)	None.
K	Cartridge tape	Mandatory when Control Table AA is required.
M	Mass storage	Mandatory when Control Table AA is required.
R	Report (printer)	None.
T	Tape (reel)	Mandatory when Control Table AA is required.
U	Unknown device	See <a href="#">When Is Device Code U Assigned?</a> for information about conditions leading to a U device code.
V	VIO (virtual I/O)	None.
X	Dummy (null device)	JOB/SCAN assigns a device code of DUMMY, if the DD statement contains a DUMMY positional parameter or the parameter DSNAME=NULLFILE has been coded. A DUMMY device code is associated for DD statements that contain a DDNAME=DD Name parameter, and the forward reference is not satisfied following IBM processing conventions.
Z	User-defined device code	JOB/SCAN allows definition of nine additional device codes.

## When Is Device Code U Assigned?

A file will be flagged with device code U (unknown) if:

- The file is presumed to be cataloged but was not in the catalog at the time of JOB/SCAN execution.
- No unit was specified for a new dataset and the dataset is not SMS-managed.
- Jobs were documented out of sequence and the file is referenced before it is actually created.
- Control Table AA is required and the unit is not defined with a table entry.
- Control Table AA is not required, but JOB/SCAN was unable to derive a device code from the catalog or the IBM Unit Verification Routine (IEFEB4UV).

## Format

Column	Entry
6-7	AA
	One of these (left-justified):
8-20	<ul style="list-style-type: none"> <li>• Device type (such as 3350)</li> <li>• Group name (such as TAPE)</li> <li>• UNIT address (such as 181)</li> </ul>
21	Standard device code (See <a href="#">Device Codes (DVC)</a> ).

## Example

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....
AATAPE          T          DEFAULT UNIT NAME FOR TAPE (REEL)
AASYSDAV        M          DEFAULT UNIT NAME FOR MASS STORAGE
AADISK          D          DEFAULT UNIT NAME FOR DASD
AA3480          K          DEFAULT UNIT NAME FOR CARTRIDGE TAPE
AA3330          D
AA330-1         D
AA3350          D
AA3380          D
AASYSDA        D
AATP6250        T
AA3330V        M
AAVIO           V

```

## Control Table AB: Control Card DD Name

Control Table AB identifies specific DD names that reference control card libraries. (See [Control Table AQ: Control Card Dataset Names](#) for information on specifying control card dataset names.) A dataset name found in Control Table CG is not considered to be a Control Card input library and JOB/SCAN does not expand it in the SJL even though it found an entry for the DD name in the AB Control Table. Use Control Table CG to restrict the expansion of datasets containing sensitive data.

## Referenced Run Option: Access Control Cards (ACC)

If the Access Control Cards Run Option is set to YES, JOB/SCAN searches to see if a Control Table AB entry is coded for each DD statement with the one of these formats:

- 1) //ddname DD \*
- 2) //ddname DD DATA
- 3) //ddname DD DSN=dsname (member) , . . . . .

If a Control Table AB entry for that DD Name is found, the contents of the member are accessed and printed in selected reports.

## Entries for the DB2 Interface

If you need to interface with batch DB2 commands, Control Table AB entries are required. For example:

```
ABSYSSTIN  IKJEFT01
ABSYSSTIN  IKJEFT1A
ABSYSSTIN  IKJEFT1B
```

## Format

Column	Entry
6-7	AB
	One of these (left-justified):
8-20	<ul style="list-style-type: none"> <li>• ALL</li> <li>• DD Name</li> </ul>
	One of these (left-justified):
21-50	<ul style="list-style-type: none"> <li>• ALL (access control cards for all programs with the DD Name specified).</li> <li>• Program name (limits access of control cards to uses by specific programs).</li> </ul>

## Example

```
.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....
ABALL      ALL      (use for all ddnames)
ABSYSIN    ALL      (use for all pgms)
ABCNTLIN   PRINTIT (PRINTIT pgm only)
```

## Control Table AC: Input/Output DD Names

Control Table AC entries contain DD names, each with a type indicator. The indicator instructs JOB/SCAN to treat files with that DD name in a particular manner, such as input or output. Use the AC Control Table entries to identify DD names where disposition does not indicate the INPUT/OUTPUT status of the dataset. An optional parameter, the program name that is associated with the DD name, can also be specified.

## Referenced Run Option: Catalog Access (CAC)

The PDSMEM parameter of the Catalog Access Run Option specifies whether or not JOB/SCAN is to validate partitioned dataset member names.

## Indicator Types

Indicator	Meaning
I (Input)	Treat the file with this DD Name as INPUT.
O (Output)	Treat the file with this DD Name as OUTPUT.

## Format

Column	Entry
6-7	AC
8-20	DD Name (left-justified)
21	Either I (Input) or O (Output)
22-29	The program name to be associated with this DD name. This is an optional parameter. If a program name is specified, the indicator will only apply to the DD name if that program is executing. If not specified, the indicator will apply to all occurrences of the DD name.

## Example

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....
  ACSYSIN          I
  ACSORTOUT        O
  ACSYSLMOD        O
  ACSYSUT2         OIEBGENER

```

## Control Table AD: Utility Programs

Control Table AD tells JOB/SCAN that a particular program requires special processing. The program name and type of processing associated with the statements must be identified with each entry. A dataset name found in Control Table CG is not considered to be a Control Card input library and JOB/SCAN does not process it even though it found an entry for the DD name in the AD Control Table

### Referenced Run Option: Program Product Interface (PPI)

The PPI Run Option is associated with Control Table AD. PPI specifies special programs JOB/SCAN can interface with. See also [Control Table AP: PARM Programs](#).

### Entries for the IMS and IDCAMS Interfaces

Both the IMS and IDCAMS interfaces require a Control Table AD entry. For this reason, JSOPT02 is precoded for those interfaces. Do not remove these entries from JSOPT02 if interfacing with IMS or IDCAMS.

The precoded IDCAMS entry is:

```
ADIDCAMS    UASYSIN
```

The precoded IMS entry is:

```
ADDFSRRC00 UR
```



Support for the DFHDRP IMS driver is indicated with a Control Table AD entry in cases where the parameters in the PARM field are keyword-driven.

## Format

Column	Entry
6-7	AD
8-15	Program requiring special processing (left-justified).
21	U (always required).
	One of these:
	<ul style="list-style-type: none"> <li>A (IDCAMS)</li> <li>P (IEHPROGM)</li> </ul>
22	<ul style="list-style-type: none"> <li>R (IMS)</li> <li>X (an ALTPGM such as SAS, FOCUS, RAMIS)</li> <li>U (User) Will cause processing at AD entry for PARM PGM=NAME. (See <a href="#">Control Table AP: PARM Programs</a>)</li> </ul>
	One of these (left-justified):
23-30	<ul style="list-style-type: none"> <li>DD name: Refers to the dataset where control statements reside.</li> <li>K: Indicates that the parameters in the parameter field are keyword-based instead of positional for purposes of substituting program names. (Also see <a href="#">Control Table AP: PARM Programs</a>)</li> </ul>

## Example

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....
ADIDCAMS          UASYSIN          IDCAMS
ADIEHPROGM        UPSYSIN          IEHPROGM UTILITY
ADDFSRRC00        UR              IMS UTILITY
ADDFHDRP          URK             IMS UTILITY
ADSASLPA          UXSYSIN
ADMYPROG          UXMYDD
ADUSERPGM         UU

```

## Control Table AM: Tape Management Locations (Out-of-Area codes)

Control Table AM contains a list of Tape Management Storage Locations to be considered as on-site.





The RMM Stor Locations, CA-DYAN/TLMS, or TMS out-of-area codes must match the code specified. Referenced Run Option: Program Product Interface (PPI)

The Program Product Interface (PPI) Run Option specifies special programs to interface with JOB/SCAN:

- If CA1 or CA1R5 is specified, Control Table AM is searched and, for any out-of-area codes listed, no message DSS8370 is issued. If RMM is specified, the AM Table is searched, and for any storage or loan locations found, no DSS5319 message is issued.
- If TLMS is specified, the AM Control Table is searched, and for any volume location code found, no DSS8395 message is issued.

### Format 1

Enter Control Table AM data for TMS, as follows:

Column	Entry
6-7	AM
8-20	TMS out-of-area code.



If leading blanks are contained in the TMS out-of-area code, they must also be coded in the Control Table AM entry. See examples AM LIB and AM VMS.

### Example

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....
      AM LIB
      AMLIBR
      AM VMS

```

### Format 2

Enter Control Table AM data for RMM, as follows:

Column	Entry
6-7	AM
8-15	RMM Stor Location or Loan Location



The Stor or Loan location to be considered as on-site might be in the RMM LOCDEF with a HOME value or a Remote site which has a tape drive.

**Example**

```
.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....
      AMDISTANT
      AMLOANLOC
```

**Format 3**

Enter Control Table AM data for CA-DYAN/TLMS as follows:

Column	Entry
6-7	AM
8-9	Volume Location Code

**Example**

```
.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....
      AMOS
      AMRL
```

**Control Table AN: Procedures Excluded from Validation**

Control Table AN contains a list of procedures to be excluded from JOB/SCAN validation. The Structured JCL Listing is not affected. Referenced Run Option: Use PROC Exclude Table (UPCX). The Use PROC Exclude Table Run Option tells JOB/SCAN whether to search Control Table AN. If this option is set to YES, JOB/SCAN excludes from validation all those PROCs listed in Control Table AN.

**Format**

Column	Entry
6-7	AN
8-20	PROC name (left-justified)

**Example**

```
.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....
      ANCOMMPROC
```

## Control Table AP: PARM Programs

Control Table AP specifies program names for which a substitute program name should be derived from a positional parameter in the PARM field. This capability is typically used for IMS and utility programs. (See the [Program Product Interface \(PPI\)](#) for information on IMS.)

JOB/SCAN derives the program name substitution from the PARM statement that applies to the step. JOB/SCAN searches Control Table AP for each program name. If a program name from an EXEC statement matches a program name from Control Table AP, the substitute program name is extracted from the PARM list.



If the parameter that specifies the program name is a keyword parameter (for example, PARM='PGM=ABC'), use Control Table AD, specifying a K in column 23.

### Format

Column	Entry
6-7	AP
8-20	Program name (left-justified)
21	Blank or expression number denoting position of a program in the PARM field; default is 1. If the field contains a 1 or a blank, the first expression from the PARM field replaces the program name. If the field contains 2 through 9, the specified field is used.
22-26	Blank spaces or the word BLANK. If the word BLANK is specified, blank spaces are used as delimiters for the PARM field strings. If the word BLANK is not specified, commas are used as the delimiters for the PARM field strings.
27-29	Reserved.
30-37	DD NAME (left-justified). This is an optional parameter. If specified, the PDS library associated with the DD NAME will be allocated, the PARM field will be considered a member name in that library, and the library will be checked to determine whether it contains that member.



If specified, the DD NAME must be allocated in the FILE ALLOCATIONS section of your SJJSPARM options member. Examples

If this is the JCL statement:

```
//STEP01 EXEC PGM=DFSRRRC00,PARM='DLI,APUPD,APPSB01'
```

and this the table entry:

```
.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....
```

APDFSRR00 2

then the existence of both APUPD and DFSRR00 is validated.

If this is the JCL statement:

```
//STEP01 EXEC PGM=APPGM, PARM='UPDPGM, XYZ, C'
```

and this the table entry:

```
.....1.....2.....3.....4.....5.....6.....
      APAPPGM
```

then the program name used for all program references is the first parameter in the PARM field (UPDPGM).

## Control Table AQ: Control Card Dataset Names

Control Table AQ designates PDS libraries that contain control cards. Control Table CG entries override Control Table AQ entries.

### Referenced Run Option: Access Control Cards (ACC)

The ACC Run Option directs JOB/SCAN to read members within the job stream that contain control cards and print them in the Structured JCL Listing. If this option is set to YES, control cards are included in the reports. In the case of DB2 and IDCAMS control cards; they are validated by JOB/SCAN if the DB2 and IDCAMS parameters are turned on in the Program Product Interface Run Option.

### How JOB/SCAN Processes the Control Table

When a DD DSN=dsname (member) statement is encountered in the JCL, JOB/SCAN searches Control Table AQ for that dataset name. If it is found, the member is read; validation is performed on DB2 and IDCAMS control statements (if the appropriate options are turned on); and the control cards are printed in the Structured JCL Listing report.

### Format

Column	Entry
6-7	AQ
8-51	Dataset (left-justified)

### Example

```
.....1.....2.....3.....4.....5.....6.....
      AQSYS1.PARMLIB
```

## Control Table AR: Return Code Simulation

Control Table AR allows JOB/SCAN to simulate non-zero return codes in the JCL being processed. Information coded in Control Table AR, such as JOBNAME, JOBSTEP, and STEPNAME, is used by JOB/SCAN to assign an abend or return code

other than zero to a program step or to indicate that the step has abended.

### Referenced Run Option: Processing Specifications (PSS)

The Processing Specifications Run Option controls the effect of the COND parameter during execution. If the PROCESSING SPECIFICATIONS=CONDCODE, the COND= parameter of each step is evaluated. Each step is assigned a return code of 0 unless a Control Table AR entry is specified, in which case simulation of non-zero return codes is performed.

#### Format

Column	Entry
6-7	AR
	job.jobstep.stepname.return code (left-justified):
	<ul style="list-style-type: none"> <li>• job: 1- to 8-character job name if a specific job; otherwise * for any/all jobs.</li> <li>• jobstep: 1- to 8-character procedure name if a specific procedure; otherwise, * for all procedures.</li> </ul>
8-51	<ul style="list-style-type: none"> <li>• stepname: 1- to 8-character job step name, or PROC step name if within a PROC.</li> <li>• return code: Either a 1- to 4-digit numeric value from 0 to 4095, or the keyword ABEND</li> <li>• ABEND=Udddd: where dddd is a numeric value between 0000 and 4095, and must contain leading zeros.</li> </ul>

#### Example

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+...
ARJOB1.PROC3.STEP5,12
AR*.GLPROC04.GLPS15,4
AR*.*.STEP4,ABEND
AR*.STEP1.ENDIT,ABEND=U0016
AR*.*.SUMMARY,ABEND=S0C7
ARJOBGL.STEP7.BACKUP,ABEND=4095

```

## Control Table AS: Nonstandard JES Statements

Control Table AS identifies any site-specific, non-IBM-standard JES statements and parameters to prevent JOB/SCAN from issuing erroneous error messages. An entry in Control Table AS is required for each nonstandard statement type and parameter. Each JES statement is passed to the standards program as an 80-column card image. Each parameter is parsed and passed, along with its value, to the standards program in a table format. In this way, users can perform standards validation on nonstandard statements and parameters, if desired.



If F is coded in Parameter type, no parsing of the parameters is performed.

The SSS Run Option tells JOB/SCAN whether you are using JES2 or JES3.

## Format

Column	Entry
6-7	AS
8-15	Statement name (nonstandard JES2 statement, left-justified).
8-17	Statement name (nonstandard JES3 statement, left-justified).
	Parameter type:
21	P for positional K for keyword F for freeform (no validation of parameters is performed)
22-31	Parameter name (name of nonstandard parameter)

## Examples

JES2 statement examples:

```
/*USRSTMT USRPARM=(OUR, INHOUSE, VALUES)
/*JOBPARM FORMS=ART, COLOR=BLUE
```

JES2 table entries:

```
.....1.....2.....3.....4.....+..
ASUSRSTMT      KUSRPARM
ASJOBPARM      KCOLOR
```

JES2 statement example using Freeform (no parsing to be done). JES2 freeform statement example:

```
/*NEWSTHE FOLLOWING COMMENTS WILL APPEAR
/*NEWSAS TEXT THAT IS PRINTED ON THE BANNER
/*NEWSPAGE OF THE OUTPUT REPORTS WHICH ARE
/*NEWSUSED FOR ROUTING INFORMATION, ETC.
```

JES3 statement examples:

```
//*FORMAT TSO,TERM=USER1
//*FORMAT PR,DDNAME=,DEST=RMT5,TERM=USER1
```

JES3 table entries:

```
.....1.....2.....3.....4.....+..
ASFORMAT      PTSO
ASFORMAT      KTERM
```

## Control Table AT: Nonstandard OUTPUT Keywords

Control Table AT can be used to identify any site-specific nonstandard OUTPUT JCL keywords to prevent JOB/SCAN from issuing erroneous error messages. For example, to allow this statement:

```
//OUT1 OUTPUT WDAY=3
```

include an ATWDAY entry in Control Table AT.

### Format

Column	Entry
6-7	AT
8-20	Keyword (left-justified)

### Example

```
.....+.....1.....+.....2.....+.....3.....+.....4.....+...
      ATWDAY
```

## Control Table AU: Job Steps to Ignore

Control Table AU contains Job steps to be ignored by JOB/SCAN. If a step name matches an entry in Control Table AU, no validation is performed. JOB/SCAN will ignore any files created or deleted by a step listed in Control Table AU. Referenced Run Option: Processing Specifications (PSS)

The Processing Specifications Run Option tells JOB/SCAN whether to exclude program steps identified by Control Table AU entries. If the option is set to CONDCODE, the program steps are excluded. (See also [Control Table AR: Return Code Simulation](#) for information on condition code simulation.)

### Format

Column	Entry
6-7	AU
	Job steps to be ignored as follows:
8-51	<ul style="list-style-type: none"> <li>• job.proc.program</li> <li>• job.(jobstep).program</li> <li>• job.proc.(procstep)</li> <li>• job.(jobstep).(procstep)</li> </ul>



Parentheses are used when specifying the step name, but not when specifying the actual PROC or PGM name.

### Example

```
.....+.....1.....+.....2.....+.....3.....+.....4.....+..
AUJOB1.PROC3.(STEP4)
```

## Control Table AW: Suppress JCL Error Messages

Control Table AW allows the user to change the severity level that JOB/SCAN automatically assigns to JCL error messages. Control Table AW also allows the user to suppress JCL error messages for all occurrences or for specific programs and DD names. Referenced Run Option: Use Error Table (UET)

If the Use Error Table Run Option is set to YES, JOB/SCAN searches Control Table AW to determine if any error messages issued on the JCL statement are listed. If the error number is found in Control Table AW, the severity is changed to the level indicated or is optionally suppressed.

### Changing Severity Level Codes

#### *To change the severity level code for a given error message*

1. Type the number of the error message needing its severity level changed, starting in column 8 of Control Table AW. (For a list of error messages and their codes, access member ERRORMSG from the SJSSAMP library.)
2. Type the new severity level code in columns 21 through 22 of Control Table AW. Severity level codes are shown in this table

Code	Level
0	Advisory
1-4	Warning
5-19	Errors
20-nn	Catastrophic (causes JOB/SCAN to stop processing).

### Suppressing Severity Level Codes for All Occurrences

#### *To suppress an error message for all occurrences*

1. Type the error message number to be suppressed, starting in column 8 of Control Table AW.
2. Either type non-numeric values or leave blanks in columns 21 through 22 of Control Table AW.

### Suppressing Severity Level Codes for a Specific Occurrence

#### *To suppress an error message for a specific occurrence*

1. Follow the steps in [Suppressing Severity Level Codes for All Occurrences](#).



2. Type a program name in columns 24 through 31, and/or a DD Name in columns 33 through 40. When error messages are suppressed, a count of the suppressed errors is included in the Error Summary at the end of the Structured JCL Listing.

### Format

Column	Entry
6-7	AW
8-14	JOB/SCAN error number such as DSS5110
15-20	Reserved.
	Optional. If specified, either:
21-22	<ul style="list-style-type: none"> <li>• A numeric severity code to change the severity level.</li> <li>• A non-numeric value or blanks to suppress the message.</li> </ul>
23	Reserved.
24-31	Optional program name (suppress message for that program only).
32	Reserved.
33-40	Optional DD Name (suppress message for that DD Name only).
21-40	Blanks in the last two fields to suppress or alter all occurrences of this error message.

### Example

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+..
      AWDSS5110      XX IDCAMS      IDCUT1
      AWDSS5110      XX IDCAMS      IDCUT2

```

## Control Table AX: Nonstandard Parameters for DD Statements

Control Table AX can be used to identify any site-specific nonstandard DD statement keywords to prevent JOB/SCAN from issuing erroneous error messages. For example, if a special DJDE DD statement keyword was defined at the user site for a Xerox 9700 printer, code an AXDJDE entry in Control Table AX.

### Format

Column	Entry
6-7	AX
8-20	Keyword (left-justified)

**Example**

```
.....1.....2.....3.....4.....
      AXDJDE
```

**Control Table AY: Nonstandard Parameters for EXEC Statements**

Control Table AY can be used to identify any site-specific nonstandard EXEC statement keywords to prevent JOB/SCAN from issuing erroneous error messages. For example, if the site has a nonstandard EXEC parameter called NEWPARM, add an AYNEWPARM entry to the table. There is no need to create AY table entries for symbolic parameters used in PROCs.

**Format**

Column	Entry
6-7	AY
8-20	Keyword (left-justified)

**Example**

```
.....1.....2.....3.....4.....
      AYNEWPARM
```

**Control Table AZ: Nonstandard Parameters for Job Statements**

Control Table AZ can be used to identify any site-specific nonstandard Job statement keywords to prevent JOB/SCAN from issuing erroneous error messages. For example, if the site has a nonstandard Job parameter called NEWPARM, add an AZNEWPARM entry to the table.

**Format**

Column	Entry
6-7	AZ
8-20	Keyword (left-justified)

**Example**

```
.....1.....2.....3.....4.....
      AZNEWPARM
```

## Control Table BA: Reformatting DD Statements

Control Table BA defines the relative line number and relative position on which specific DD statement parameters should be placed. For example, specify that DISP should appear on line 03 of every DD statement as the second parameter. Any parameters not defined by Control Table BA are formatted according to default values.

### Control Table Entries

All IBM parameters, and any additional user-defined parameters and symbolics, can be coded in Control Table BA. If the JCL contains DD statement parameters not specified in Control Table BA or in the default list, those parameters are placed at the end of the rest of the parameters.



- If multiple parameters (for example, DSN and UNIT) are specified for the same line and position, they are placed in their original order.
- To turn on DD statement parameter reformatting, code at least one BA control table entry. To prohibit reformatting of DD statements, remove all BA Control Tables from the SJJSPARM members and if using batch, from JSMVS member SYSIN.

### Alphanumeric Sorting of DD Parameters

Because there might be many DD statement parameters not defined in Control Table BA either explicitly or by default, you can force alphanumeric sorting of these parameters by coding a BA00 entry. This entry causes all parameters not defined the BA table to be sorted alphanumerically and written one parameter per line in the reformatted JCL after the explicitly defined parameters.

### Default Values

Relative Line No./ Position No.	Parameter	Relative Line No./ Position No.	Parameter	Relative Line No./ Position No.	Parameter
01 00	*	02 99	DLM	03 99	OUTLIM
01 00	DATA	02 99	FCB	03 99	AFF
01 00	DUMMY	02 99	MGMTCLAS	03 99	SPLIT
01 00	DYNAM	02 99	STORCLAS	04 99	SPACE
01 00	&&	02 99	UCS	05 99	EXPDT
01 99	DDNAME	02 99	QNAME	05 99	LABEL
01 99	DSNAME	02 99	TERM	05 99	RETPD

Relative Line No./ Position No.	Parameter	Relative Line No./ Position No.	Parameter	Relative Line No./ Position No.	Parameter
01 99	DSN	02 99	SUBALLOC	06 99	VOLUME
01 99	LIKE	02 99	FREE	06 99	VOL
01 99	REFDD	02 99	CHARS	19 99	PROTECT
01 99	SECMODEL	02 99	BURST	19 99	KEYOFF
01 99	SYSOUT	02 99	CHKPT	19 99	RECORG
02 99	AVGREC	02 99	DSID	19 99	KEYLEN
02 99	DATACLAS	02 99	FLASH	19 99	LRECL
02 99	DISP	02 99	MODIFY	19 99	RECFM
02 99	COPIES	02 99	OUTPUT	20 99	DCB
02 99	DEST	03 99	MSVGP	20 99	AMP
02 99	HOLD	03 99	UNIT	21 99	SUBSYS

**Format**

Column	Entry
6-7	BA
8-9	Two digits representing the relative line number within each DD statement. For numbers under 10, use a leading zero (example, 02).
10-12	(optional) A comma followed by two digits representing the relative position number within each line. For numbers under 10, use a leading zero (for example, 02 ).
10-17 or 13-20	DD parameter keyword

**Examples**

The example shows extensive reformatting for DD parameters.

```
.....+.....1.....+.....2.....+.....3.....+.....4.....+..
      BA03LABEL
```

```
BA04UNIT
BA05DISP
BA05HOLD
BA06,01COPIES
BA06,02DEST
BA02,01VOLUME
BA02,01VOL
```

This example shows control tables for reformatting two DD parameters and customized columnar starting positions.

```
.....1.....+.....2.....+.....3.....+.....4.....+..
BA04UNIT
BA05DISP
BC                               11 15 15
```

## Control Table BB: Reformatting EXEC Statements

Control Table BB defines the relative line number and relative position on which specific EXEC statement parameters should be placed. For example, specify that PARM should appear on line 03 of every EXEC statement. Any parameters not defined by Control Table BB are formatted according to default values.

### How Reformatter Handles Control Table BB Entries

All IBM parameters, and any additional user-defined parameters and symbolics, can be coded in Control Table BB. If the JCL contains EXEC statement parameters not specified in Control Table BB or in the default list, those parameters are placed at the end of the rest of the parameters. If more than one parameter (for example, DSN and UNIT) are specified for the same line and position, they are placed in their original order.



- Do not use an ampersand (&) when coding symbolic parameters for Control Table BB.
- To turn on EXEC statement parameter reformatting, code at least one BB table entry. To prohibit reformatting of EXEC statements, remove any Control Table BB statements from the SJJSPARM members and if using batch, from JSMVS member SYSIN.

### Alphanumeric Sorting of EXEC Parameters

Because there might be many EXEC statement parameters not defined in Control Table BB either explicitly or by default, you can force alphanumeric sorting of these parameters by coding a BB00 entry. This entry causes all parameters not defined the BB table to be sorted alphanumerically and written one parameter per line in the reformatted JCL after those explicitly defined parameters.

### Default Values

Relative Line No./ Position No.	Parameter	Relative Line No./ Position No.	Parameter	Relative Line No./ Position No.	Parameter
01 00	PGM	02 99	REGION	04 99	ACCT
01 00	PROC	03 99	PARM	04 99	ADDRSPC

Relative Line No./ Position No.	Parameter	Relative Line No./ Position No.	Parameter	Relative Line No./ Position No.	Parameter
01 99	COND	04 99	RD	10 99	DPRTY
02 99	TIME	04 99	PERFORM	11 99	DYNAMNBR

## Format

Column	Entry
6-7	BB
8-9	Two digits representing the relative line number within each EXEC statement. For numbers under 10, use a leading zero (for example, 02).
10-12	(optional) A comma followed by two digits representing the relative position number within each line. For numbers less than 10, use a leading zero (for example, 02).
10-17 or 13-20	EXEC parameter keyword.

## Example

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+..
BB02PARM
BB03SITEPARM
BB05ACCT
BB05REGION

```

This example shows control tables for reformatting two parameters and customized columnar starting positions.

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+..
BB05ACCT
BB05REGION
BC                11 15 15

```

## Control Table BC: Additional Reformatter Options

A Control Table BC entry:

- Alters or removes the line sequence number in positions 73 through 80 of reformatted JCL statements.
- Customizes the starting position for the operation field.
- Customizes the starting position for the first parameter on the operation field line.
- Customizes the starting position for continuation lines.
- Restructures comment fields and reposition comment statements.
- Removes any comment fields from the JCL.


Each of these tasks is performed on a JCL statement only if the corresponding Reformatter Table entry is also present. For example, to get comment field reformatting on a JOB statement, code both a BC table entry and a BE table entry.

## Default Values

This table describes the default values for Control Table BC, which are diagrammed here:

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5..
      BC          112 18 16 BLM  12 16 16 12 16 16
  
```

Control Table Default Value	Column Numbers	Meaning of Default Value
BC	6-7	Table name.
	8-12	Start renumbering with current line number.
,	13	Required comma only for values in columns 8-12 and 14-18.
	14-18	Increment by current increment.
1	20	Select sentence mode for comment fields.
12	21-22	Start the operation field for EXEC, PROC, PEND, SET, JCLLIB, and INCLUDE statements in position 12.
18	24-25	Start the first parameter on the operation field line for EXEC, PROC, PEND, SET, JCLLIB, and INCLUDE statements in position 18.
16	27-28	Start the parameters on continuation lines for EXEC, PROC, PEND, SET, and JCLLIB statements in position 16.
		<div style="border: 1px solid purple; padding: 5px; display: inline-block;">  There is no continuation for INCLUDE statements.         </div>
B	30	Move the comment line to the bottom of the statement.
L	31	Associate comment field with last parameter on line.
M	32	Keep comment in original placement.
12	36-37	Start the operation field for DD and OUTPUT statements in position 12.
16	39-40	Start the first parameter on the operation field line for DD and OUTPUT statements in position 16.
16	42-43	Start the parameters on continuation lines for DD and OUTPUT statements in position 16.

Control Table Default Value	Column Numbers	Meaning of Default Value
12	45-46	Start the operation field for JOB statements in position 12.
16	48-49	Start the first parameter on the operation field line for JOB statements in position 16.
16	51-52	Start the parameters on continuation lines for JOB statements in position 16.




- Reformatter reformats all parameter positions not defined in Control Table BC according to the default values.
- Reformatter restructures comment fields in sentence mode when comment restructuring is not specified by a BC table or a Reformatter command.

### Permissible Values for Control Table BC

This table describes the permissible values for the BC Control Table.

Column	Entry
6-7	BC.
8-12	Five digits representing starting line sequence number to use in renumbering. For numbers under 10000, use leading zeros (example, 00100). To delete sequence numbers, use 00000. IF WCO=YES, and the sequence number field is blank already, the member will not be written, assuming no other changes have been made.
13	A required comma only for values in columns 8-12 and 14-18.
14-18	Five digits representing increment value to use in renumbering. For numbers under 10000, use leading zeros (example, 00100). To delete sequence numbers, use 00000.
20	'1' Select sentence option for processing comments (default). '2' Select affinity option for processing comments. '3' Select template option for processing comments. 'N' Removes comment fields on all JCL statements. (This specification does not remove <code>/**</code> comment statements.)
21-22	A number from 04 to 60 specifying position in which to start the operation field of EXEC, PROC, PEND, SET, JCLLIB, and INCLUDE statements.



Column	Entry
24-25	A number from 04 to 60 specifying position in which to start the first parameter on operation field line of all EXEC, PROC, PEND, SET, JCLLIB, and INCLUDE statements.
27-28	A number from 04 to 16 specifying position in which to start the first parameter on continuation lines of all EXEC, PROC, PEND, SET, and JCLLIB statements.
	<div style="border: 1px solid black; padding: 5px;">  There is no continuation for INCLUDE statements. </div>
30	<p>Affinity option processing imbedded comment lines:</p> <p>B Move imbedded comment line to the bottom of a statement.</p> <p>T Move imbedded comment to the top of a statement.</p> <p>L Leave comment in the same position (don't move anything).</p>
31	<p>Affinity options for parameter association of comment fields:</p> <p>L (default) Associate with the last parameter on the line.</p> <p>F Associate with the first parameter on the line.</p> <p>A Associate with all parameters on the line.</p> <p>Template options:</p> <p>1 Eliminate resulting blank comment lines.</p>
32	<p>Affinity options for column placement of comment fields:</p> <p>M Maintain original column placement.</p> <p>S Specify a column where to start the comment field.</p> <p>R Right justify the comment field to column 71.</p> <p>L Left justify the comment field 1 byte after the parameter.</p>
33-34	Affinity option for specific column placement (when column 32=S): nn number from 04 to 72 representing starting column to place comment field.
36-37	A number from 04 to 60 specifying position in which to start operation field of all DD and OUTPUT statements.
39-40	A number from 04 to 60 specifying position in which to start the first parameter on the operation field line of all DD and OUTPUT statements.
42-43	A number from 04 to 16 specifying position in which to start the first parameter on the continuation lines of all DD and OUTPUT statements.
45-46	A number from 04 to 60 specifying position in which to start the operation field of all JOB statements.
48-49	A number from 04 to 60 specifying position in which to start the first parameter on the operation

Column	Entry
	field line of all JOB statements.
51-52	A number from 04 to 16 specifying position in which to start the first parameter on the continuation line of all JOB statements.

**Example 1 - Renumber, Remove Comment Fields, Set the Continuation Column**

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+...
      BC00020,00020 N          10
    
```


Control Table Value	Column Numbers	Meaning
BC	6-7	Table name.
00020	8-12	Start renumbering at line 20.
00020	14-18	Increment renumbering by 20.
N	20	Remove all comment fields from the JCL.
10	27-28	Start the first parameter on continuation lines on EXEC and PROC statements in position 10.

**Example 2 - Renumber, Remove Comment Fields, Set the Start Positions**

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+...
      BC00100,00050 312 17 13      13 17 13
    
```

Control Table Value	Column Numbers	Meaning
BC	6-7	Table name.
00100	8-12	Start renumbering at line 100.
00050	14-18	Increment renumbering by 50.
3	20	Use template mode for comment fields.
12	21-22	Start the operation field for EXEC, PROC, PEND, SET, JCLLIB, and INCLUDE statements in position 12.

Control Table Value	Column Numbers	Meaning
17	24-25	Start the first parameter on the operation field line for EXEC, PROC, PEND, SET, JCLLIB, and INCLUDE statements in position 17.
13	27-28	Start the parameters on continuation lines for EXEC, PROC, PEND, SET, and JCLLIB statements in position 13. <div style="border: 1px solid purple; padding: 5px; margin-top: 10px;">  There is no continuation for INCLUDE statements. </div>
13	36-37	Start the operation field for DD and OUTPUT statements in position 13.
17	39-40	Start the first parameter on the operation field line for DD and OUTPUT statements in position 17.
13	42-43	Start the parameters on continuation lines for DD and OUTPUT statements in position 13.

### Example 3 - Remove All Sequence Numbers

```
.....+.....1.....+.....2.....+.....3.....+.....4.....+...
      BC00000,00000
```

## Control Table BD: Reformatting PROC Statements

Control Table BD defines the relative line number and relative position on which specific PROC statement parameters should be placed. If no entries are made in this table, PROC statements are not reformatted. There are no default values for Control Table BD.

### How Reformatter Handles Control Table BD Entries

Any number of user-defined parameters and symbolics can be coded in Control Table BD. If the JCL contains PROC statement parameters not specified in Control Table BD, those parameters are placed at the end of the rest of the specified parameters. If more than one parameter (for example, DSN and UNIT) are specified for the same line and position, they are placed in their original order.



- Do not use an ampersand (&) when coding symbolic parameters for Control Table BD.
- To turn on PROC statement parameter reformatting, code at least one BD Control Table entry.



- To prohibit reformatting of PROC statements, remove all BD Control Tables from the SJJSPARM members and if using batch, from JSMVS member SYSIN.

## Alphanumeric Sorting PROC Parameters

Because there might be many PROC statement parameters not defined in Control Table BD either explicitly or by default, you can force alphanumeric sorting of these parameters by coding a BD00 entry. This entry causes all parameters not defined the BD table to be sorted alphanumerically and written one parameter per line in the reformatted JCL after those explicitly defined parameters.

### Format

Column	Entry
6-7	BD
8-9	Two digits representing the relative line number within each PROC statement. For numbers under 10, use a leading zero (example, 02).
10-12	(optional) A comma followed by two digits representing the relative position number within each line. For numbers under 10, use a leading zero (for example, 02).
10-17 or 13-20	PROC parameter keyword.

### Example

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+...
BD01DATEPARM
BD02RMTSITE
BD02TAPEVOL

```

This example shows control tables for reformatting two parameters and customized columnar starting positions.

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+..
BD01DATEPARM
BD02RMTSITE      11 15 15
BC

```

## Control Table BE: Reformatting JOB Statements

Control Table BE defines the relative line number and relative position on which specific JOB statement parameters should be placed. For example, specify that MSGCLASS should appear on line 03 of every JOB statement. Any parameters not defined by Control Table BE are formatted according to default values.

### How Reformatter Handles Control Table BE Entries

All IBM parameters, and any additional user-defined parameters and symbolics, can be coded in Control Table BE.

If the JCL contains JOB statement parameters not specified in Control Table BE or in the default list, those parameters are placed at the end of the rest of the parameters. If more than one parameter (example, NOTIFY and USER) are specified for the same line and position, they are placed in their original order.



- To turn on JOB statement parameter reformatting, code at least one BE table entry.
- To prohibit reformatting of JOB statements, remove all Control Table BE statements from the SJJSPARM members and if using batch, from JSMVS member SYSIN.

### Sorting JOB Parameters Alphanumerically

Because there might be many JOB statement parameters not defined in Control Table BE either explicitly or by default, you can force alphanumeric sorting of these parameters by coding a BE00 entry. This entry causes all parameters not defined the BE table to be sorted alphanumerically and written one parameter per line in the reformatted JCL after those explicitly defined parameters.

### Default Values

Relative Line No./ Position No.	Parameter	Relative Line No./ Position No.	Parameter	Relative Line No./ Position No.	Parameter
01 00	ACCT	02 99	ADDRSPC	02 99	PRTY
01 01	PGMRNAME	02 99	BYTES	02 99	RD
01 99	MSGCLASS	02 99	CARDS	02 99	REGION
01 99*	MSGLEVEL	02 99	GROUP	02 99	SECLABEL
01 99	NOTIFY	02 99	LINES	03 99	COND
01 99	USER	02 99	PAGES	03 99	RESTART
01 99	CLASS	02 99	PASSWORD	03 99	TYPRUN
01 99	TIME	02 99	PERFORM		

### Format

Column	Entry
6-7	BE
8-9	Two digits representing the relative line number within each JOB statement. For numbers under 10, use a leading zero (example, 02).

Column	Entry
10-12	(optional) A comma followed by two digits representing the relative position number within each line preceded by commas. For numbers under 10, use a leading zero (for example, 02).
10-17 or 13-20	JOB parameter keyword.

**Example**

```


.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+
BC                                     11 17 15
BE02PGMRNAME
BE03,01CLASS
BE03,02MSGCLASS
    
```

## Control Table BF: Reformatting OUTPUT Statements

Control Table BF defines the relative line number and relative position on which specific OUTPUT statement parameters should be placed. For example, specify that CLASS should appear on line 03 of every OUTPUT statement.

### How Reformatter Handles Control Table BF Entries

All IBM parameters, and any additional user-defined parameters and symbolics, can be coded in Control Table BF. If the JCL contains OUTPUT statement parameters not specified in Control Table BF or in the default list, those parameters are placed at the end of the rest of the parameters. If more than one parameter (example, DEST and FORMS) are specified for the same line and position, they are placed in their original order.

 • To turn on OUTPUT statement parameter reformatting, code at least one BF table entry.

• To prohibit reformatting of SET statements, remove all Control Table BF statements from the SJJSPARM members and if using batch, from JSMVS member SYSIN.

### Sorting OUTPUT Parameters Alphanumerically

Because there might be many OUPUT statement parameters not defined in Control Table BF either explicitly or by default, you can force alphanumeric sorting of these parameters by coding a BF00 entry. This entry causes all parameters not defined the BF table to be sorted alphanumerically and written one parameter per line in the reformatted JCL after those explicitly defined parameters.

### Default Values

Relative Line No./ Position No.	Parameter	Relative Line No./ Position No.	Parameter	Relative Line No./ Position No.	Parameter
01 99	CLASS	02 99	THRESHLD	04 99	OFFSETXB

Relative Line No./ Position No.	Parameter	Relative Line No./ Position No.	Parameter	Relative Line No./ Position No.	Parameter
01 99	COMPACT	03 99	BURST	04 99	OFFSETXF
01 99	DEST	03 99	CONTROL	04 99	OFFSETYB
01 99	FCB	03 99	DATAACK	04 99	OFFSETYF
01 99	FORMDEF	03 99	DEFAULT	04 99	OUTBIN
01 99	FORMS	03 99	DPAGELBL	04 99	OVERLAYB
01 99	GROUPID	03 99	JESDS	04 99	OVERLAYF
01 99	MODIFY	03 99	PIMSG	04 99	OVFL
01 99	PAGEDEF	03 99	PRMODE	04 99	PORTNO
01 99	UCS	03 99	SYSAREA	04 99	PRTERROR
01 99	WRITER	03 99	TRC	04 99	PROPTNS
02 99	CHARS	04 99	ADDRESS	04 99	PRTQUEUE
02 99	CKPTLINE	04 99	BUILDING	04 99	RESFMT
02 99	CKPTPAGE	04 99	COLORMAP	04 99	RETAINS
02 99	CKPTSEC	04 99	COMSETUP	04 99	RETAINF
02 99	COPIES	04 99	DEPT	04 99	RETRYL
02 99	FLASH	04 99	DUPLEX	04 99	RETRYT
02 99	INDEX	04 99	FORMLEN	04 99	ROOM
02 99	LINDEX	04 99	FSSDATA	04 99	TITLE
02 99	LINECT	04 99	INTRAY	04 99	USERDATA
02 99	OUTDISP	04 99	NAME	04 99	USERLIB
02 99	PRTY	04 99	NOTIFY		

## Format

Column	Entry
6-7	BF
8-9	Two digits representing the relative line number within each OUTPUT statement. For numbers under 10, use a leading zero (example, 02).
10-12	(optional) A comma followed by two digits representing the relative position number within each line. For numbers under 10, use a leading zero (for example, 02).
10-17 or 13-20	OUTPUT statement keyword.

## Example

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+....
BC                                     11 17 15
BF02CLASS
BF03,01DEST
BF03,02FCB

```

## Control Table BG: Reformatting SET Statements

Control Table BG defines the relative line number and position on which specific SET statement parameters should be placed. If no entries are made in this table, SET statements are not reformatted. There are no default values for Control Table BG.

### How Reformatter Handles Control Table BG Entries

Any number of user-defined parameters and symbolics can be coded in Control Table BG. If the JCL contains SET statement parameters not specified in Control Table BG, those parameters are placed at the end of the rest of the specified parameters. If multiple parameters are specified for the same line and position, they are placed in their original order.



- Do not use an ampersand (&) when coding symbolic parameters for Control Table BG.
- To turn on SET statement reformatting, code at least one BG table entry.
- To prohibit reformatting of SET statements, remove all Control Table BG statements from the SJJSPARM members and if using batch, from JSMVS member SYSIN.

### Sorting SET Parameters Alphanumerically

Because there might be many SET statement parameters not defined in Control Table BG either explicitly or by default, you can force alphanumeric sorting of these parameters by coding a BG00 entry. This entry causes all parameters not defined the BG table to be sorted alphanumerically and written one parameter per line in the reformatted JCL after those explicitly defined parameters.



## Format

Column	Entry
6-7	BG
8-9	Two digits representing the relative line number within each SET statement. For numbers <10, use a leading zero (for example, 02).
10-12	(optional) A comma followed by two digits representing the relative position number within each line. For numbers <10, use a leading zero (for example, 02).
10-17 or 13-20	SET parameter keyword.

## Example

Showing line number reformatting only:

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+...
      BC          11 17 15
      BG01INHLQ

```

Showing line and position reformatting:

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+...
      BC          11 17 15
      BG01INHLQ
      BG01,02OUTHLO

```

## Control Table BH: Reformatting JCLLIB Statements

Control Table BH defines the reformatting option for the JCLLIB statement. If this table is not coded, JCLLIB statements are not reformatted. There are no default values for Control Table BH.

### How Reformatter Handles Control Table BH Entries

Because the JCLLIB statement has only one keyword, no relative line or position numbers are required on Control Table BH. The starting positions for the operation field and keyword are defined in Control Table BC. They can be changed using Control Table BC or JCL Reformatter commands.



- To turn on JCLLIB statement reformatting, code at least one BH table entry.
- To prohibit reformatting of JCLLIB statements, remove all Control Table BH statements from the SJJSPARM members and if using batch, from JSMVS member SYSIN.

## Format

Column	Entry
6-7	BH

## Example

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+...
      BC
      BH
      11 17 15

```

## Control Table BI: Reformatting INCLUDE Statements

Control Table BI defines the reformatting option for the INCLUDE statement. If no entries are made in this table, INCLUDE statements are not reformatted. There are no default values for Control Table BI.

Because the INCLUDE statement has only one keyword, no relative line or position numbers are specified. The starting positions for the operation field and keyword are defined in Control Table BC. They can be changed using Control Table BC or JCL Reformatter commands.



- To turn on INCLUDE reformatting, code at least one BI table entry.
- To prohibit reformatting of INCLUDE statements, remove all Control Table BI statements from the SJJSPARM members and if using batch, from JSMVS member SYSIN.

## Format

Column	Entry
6-7	BI

## Example

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+...
      BC
      BI
      12 18 16

```

## Control Table CB: Program Name Exit

Control Table CB allows you to specify a program name exit that will be called when a specified program name occurs on an EXEC PGM statement. Data required for the program name exit can be in the PARM field of the EXEC PGM statement, or in a control card file (80 column, fixed length records) whose DD Name is specified in the Control Table CB entry. The program name exit can return an error message to be printed on the Structured JCL Listing.

See [Program Name Exit: How It Works](#) for complete information about using program name exits.

## Referenced Run Option: Access Control Cards (ACC)

The ACC Run Option must be YES if the DD Name used in a Control Table CB entry points to a partitioned dataset and member.

You must also code a corresponding Control Table AB entry if the DD Name used in a Control Table CB entry points to a partitioned dataset and member.

### Format

Column	Required	Entry
6-7	Yes	CB
8-20	Yes	The program name coded on the EXEC PGM statement that requires calling a program name exit.
21-30	Yes	The program name exit to call.
31-40	No	The DD Name of the DD statement that identifies a file (dataset and member name) containing SYSIN control card data to pass to the program name exit.

### Example

```

-----+-----1-----+-----2-----+-----3-----+-----4-----
CBADATABASE      ADBEXIT
CBNATURAL        ADBEXIT  NATURALD
CBSUPRA          SUPEXIT  SUPRAD
  
```

## Control Table CC: VOLSER

Control Table CC specifies volume names that can appear in the VOL=SER= parameter of a DD statement or the VOLUMES parameter of an IDCAMS control statement. JOB/SCAN bypasses the volume verification for all volumes specified in Control Table CC entries.

If there is a VOL=SER=DSS801 parameter in the JCL and DSS801 does not exist, JOB/SCAN will not issue an error message if it finds a Control Table CC entry for DSS801.

### Conditions Requiring CC Entries

Control Table CC entries are required if the customer site requires bypassing the JOB/SCAN volume verification process.

### Format

Column	Entry
6-7	CC
8-13	Volume name (left-justified)

**Example**

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....
      CCSS801                                DO NOT CHECK VOL=SER=DSS801

```

**Control Table CD: DB2 Group Attachment Names**

Control Table CD contains user-specified DB2 Group Attachment Names and the associated DB2 Subsystem Name. JOB/SCAN will check for an active associated subsystem when a Group Attachment Name is specified. Group Attachment Names specified in the Control Table CD will eliminate the DSS8104 message if the associated subsystem is active.

**Conditions Requiring CD Entries**

Control Table CD entries are required if JCL validation requires DB2 Group Attachment Name processing.

**Format**

Column	Entry
6-7	CD
8-11	DB2 Group Attachment Name
21-24	DB2 Subsystem Name

**Example**

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....
      CDDSN                                DSN1

```

**Control Table CE: Scheduler Restart Program Names**

Control Table CE contains the names of scheduler restart programs. JOB/SCAN simulates the operation of the restart program when one of these programs is found.

**Conditions Requiring CE Entries**

Control Table CE entries are required if the customer site is using a scheduling product.

**Format**

Column	Entry
6-7	CE
8-15	Scheduler restart program name

**Example**

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....
CEU11RMS
CECTRCTR
CEEQQDELDS

```

## Control Table CF: Subsystem Dataset Disposition

Entries in the CF table tell JOB/SCAN to determine the disposition of subsystem datasets (DD statements that have a subsystem specified) from the DISP parameter. If the subsystem name from the DD statement is not found in the CF table, JOB/SCAN ignores the DISP parameter and uses the default disposition `DISP=(MOD,KEEP)`. However, when the subsystem LAM is specified in the DD statement and no corresponding CF table entry is found, JOB/SCAN uses `DISP=OLD`.



This control table is consulted only when a SUBSYS parameter is present.

### Format

Column	Entry
6-7	CF
8-11	Subsystem name (left justified)
12-20	Blank
21 and above	Comment

### Example

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----6-----+-----7--
CFSOV          EXTENDED HIGH PERFORMANCE DATA MOVER (EXHPDM)

```

## Control Table CG: Datasets That Are Not Control Card Datasets

Control Table CG contains names of Control Card datasets that are not to be expanded in the SJL or processed as utility program input. These entries override the corresponding entries in Control Tables AB, AD and AQ.

### Format

Column	Entry
6-7	CG
8-51	Dataset (left justified)

### Example

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----6-----+-----7--
CGDSSI.TEST.SJJSPARM

```

## Control Table HF: HFS Path Name DD statements

Control Table HF tells JOB/SCAN that HFS path names associated with the DD names specified in this table are not to be validated. JOB/SCAN does not validate any path names concatenated to the DD names listed in this table.

### Format

Column	Entry
6-7	HF
8-15	DD name (left justified)
16-80	Blank

### Example

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----6-----+-----7--
      HFDDNAME1
      HFDDNAME2
    
```

## Control Table Ux: Standards Facility User Table

The Standards Facility of JOB/SCAN provides the ability to set up external tables and callable functions to process the data in the tables. Entering valid site conventions using Control Table entries, as opposed to hardcoding them in the program, allows data to be changed at run time without program modifications and recompilations.

Control Table Ux entries can contain information such as lists of valid system codes, accounting parameters, DSN prefixes, etc.

### Format

Column	Entry
6	U
7	Any unique alphanumeric or national character (@, #, \$).
8-72	Up to 65 user-defined characters. Only one value per entry is permitted. It is recommended that the same format be used for each entry type.

Different Ux Control Tables can be set up for different purposes. For example, UA entries might contain valid application IDs, although UB entries might contain valid department IDs.

### Example

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....
      UAPAY                                PAYROLL
    
```

UAINV  
UAACC  
UBD/165  
UBD/209  
UBD/322

INVENTORY  
ACCOUNTING  
MARKETING DEPT.  
ENGINEERING DEPT.  
SALES DEPT.

# Scheduler Interfaces

Validating run-time JCL is essential to accurate simulation and validation. Utilizing JOB/SCAN's interfaces to schedules ensures that variables are resolved appropriately and that members are validated in the order the scheduler product schedules them.

- [CA-7 Scheduler Interface](#)
- [Handling #J Cards with the CA-7 Scheduler Interface](#)
- [Validating Jobs in Run-time Order with the CA-7 Scheduler Interface](#)
- [CA-Scheduler Interface](#)
- [CA-Driver Interface](#)
- [CONTROL-M Scheduler Interface](#)
- [DCMS Scheduler Interface](#)
- [ESP Scheduler Interface](#)
- [JOBTRAC Scheduler Interface](#)
- [TWS z/OS Scheduler Interface](#)
- [ASG-Zeke Scheduler Interface: \\$ Symbolic Parameters](#)
- [BETA42 Scheduler Interface](#)

## CA-7 Scheduler Interface

JOB/SCAN offers two interfaces to the CA-7 scheduler:

- The preprocessing interface resolves any CA-7 #J cards in the JCL.
- The batch interface to CA-7 validates scheduled production jobs in run-time order.

### Preprocessing Jobs Containing #J Cards

If jobs contain CA-7 #J cards, the cards must be preprocessed to prevent JCL errors and, optionally, to simulate an actual production run. ASG provides two sample programs in the SJSSAMP library to preprocess JCL containing #J statements for the CA-7 Scheduler. CA7J1 can process a single schedule ID, while CA7J2 can process from one to five schedule IDs. Except for the number of schedule IDs handled, the programs perform the same function. See [Selection Exit Interface](#) for detailed information about modifying the sample programs for your requirements.

### Validating Jobs in Run-time Order

If validating jobs in run-time order, a batch interface program is provided by ASG, along with the JCL necessary to run it. The interface extracts the information from CA-7 and automatically validates the jobs in their scheduled order. If CA-7 #J cards are intended to include or exclude a part of a DD statement or a certain keyword based on the schedule ID, multiple occurrences of the keyword may be present in the DD statement. JOB/SCAN's Reformatter considers any multiple occurrence of a keyword within a single DD statement as a syntax error and terminates processing of the member. An error message is generated, but the input JCL job stream remains intact.



## Handling #J Cards with the CA-7 Scheduler Interface

If using CA-7 #J control cards in the JCL, invoke the CA7J1 or CA7J2 program to:

- Prevent JCL errors caused by #J cards.
- Interpret the information on the #J cards to simulate the actual JCL as it is executed.

CA7J2 is used to pass multiple schedule IDs; CA7J1 is used for a single schedule ID.

[Guidelines for Specifying Parameters](#) and [Validating JCL that Includes CA-7 #J Statements](#) describe how to invoke and pass parameters to the sample CA7J1 and CA7J2 programs. See [Selection Exit Interface](#) for detailed information about modifying the sample programs.

### Invoking the CA7J1 Program

Invoke the CA7J1 or CA7J2 program by specifying its name and a parameter on the JCL Validation panel or in the INPUT SELECTION section of a batch job stream. For example, simulating a schedule ID of 010, a date of March 8, 1992 (68th day of 1992), and time of 17:30, would appear as:

On the JCL Validation panel:

```
Selection Exit:===> CA7J1
Parm===> 010920681730
```

In a batch job stream:

```
INPUT SELECTION
SELECTION EXIT=CA7J1,'010920681730'
```

### Guidelines for Specifying Parameters

Follow these guidelines when specifying parameters:

- Specify the parameter in IIIYYDDDDHHMMYYDDDDHHMM where:
  - III is the schedule ID to be used for this run.
  - YYDDD is the Julian beginning and ending date to be used for this job, if Julian date is used in #J cards.
  - HHMM is the hours and minutes for beginning and ending dates, in military format, if used in #J cards.
- If the selection exit to be in effect is needed for the current day only, omit the ending date from the parameter.
- The values in the Parm field are positional. If date or time fields on the #JI and #JO cards are not used, leave the date and time fields blank. However, if using a time but no date, enter zeros in the date field.
- If CA7J1 or CA7J2 encounters #JI,..., #JO,... or #JEND cards, it looks at the schedule ID and the beginning and ending dates and times on the cards and compares those values with the values specified in the Parm field. On the basis of this comparison, the program either includes or omits the cards.

## Validating Jobs in Run-time Order with the CA-7 Scheduler Interface

To validate a set of jobs accurately, it is sometimes desirable to process the jobs in run-time order. ASG provides a program and JCL to interface JOB/SCAN with the CA-7 (UCC-7) Scheduler.

### The Interface

The interface consists of a three-step job that:

- Executes the CA-7 SASSBSTR program to produce a list of jobs in run-time order.
- Executes the ASG CA7INTF program to format the job names into a list of jobs to be validated by JOB/SCAN.
- Executes JOB/SCAN to validate the jobs in run-time order.

### Establishing the Interface

#### *To establish the interface*

1. Verify that the interface program shipped by ASG is correct for your site's release of CA-7.
  - a. Print a source listing of the CA7INTF COBOL program in the SJSSAMP library.
  - b. Check the input File Description and the input record layout, and modify the COBOL code if necessary.
  - c. Compile and link the program. (To compile JOB/SCAN COBOL exits, use the ISPF edit macro CCEXIT in SJSSAMP.)
  - d. Verify that the program is creating a series of appropriate SELECT= statements, which will be used by JOB/SCAN.
  - e. Verify a Structured JCL Listing is being created for each job in the series.
2. If hardcopy is desired from validation of jobs, change the Report option in JSOPT02 for the Structured JCL Listing to =PRINT. The default is NOPRINT.
3. Modify the Interface JCL and Submit the Job
  - a. Determine the criteria for extracting job streams from the CA-7 Scheduler.
  - b. Add appropriate SYSIN control cards to the first step of the JCL contained in the CA7SCHED member of the SJSSAMP library. (See the CA-7 documentation for information on the format of the FJOB command.)

For example:

```
/LOGON password
      FJOB, JOB=*, FROM=mmddy, SPAN=nn
/LOGOFF
```

where:

password is the CA-7 password.

mmddy is the beginning date requested for the forecast (in Gregorian format).

nn is any value that is valid for CA-7.

4. Modify the rest of the CA7SCHED JCL according to the directions contained in the member.
5. Submit the job.

The figure shows sample JCL for the CA-7 Scheduler interface (in the CA7SCHED member of the SJJSSAMP library):

#### Sample CA7SCHED JCL

```
//CA7SCHED JOB (ACCT), 'PROGRAMMER NAME'
//*****
//*   INTERFACE TO CA-7 SCHEDULER
//*   REF:   SJJSSAMP (CA7SCHED)
//*
//*   NOTE: MODIFY JOB STATEMENT AND ITEMS INDICATED BY ARROWS <==
//*****
//***** GENERATE CA-7 SCHEDULE REPORT *****
//SCHEDULE EXEC PGM=SASSBSTR
//CA7CMDS   DD DSN=YOUR.CA7.CMDS,DISP=SHR           <== CA-7 CMD FILE
//BATCHIN   DD DSN=YOUR.CA7.BATCH,DISP=SHR         <== CA-7 INPUT
//BATCHOUT  DD DSN=YOUR.CA7.BATCHOUT,DISP=SHR      <== CA-7 OUTPUT
//SYSPRINT  DD DSN=&&CA7RPT,DISP=(,PASS),
//          SPACE=(TRK,(30,10),RLSE),UNIT=SYSDA,   <== VALID UNIT
//          DCB=(RECFM=FBA,LRECL=133,BLKSIZE=6118)
//*          => CHANGE MMDDYY AND HHMM VALUES BELOW <=====
//SYSIN     DD DATA,DLM=ZZ
//LOGON     PASSWORD
//JOB, SYS=GG, FROM=(MMDDYY,HHMM), TO=(MMDDYY,HHMM)
//LOGOFF
//ZZ
//SYSUDUMP  DD SYSOUT=*
//*
//***** GENERATE SELECTION STATEMENTS FOR THE SCHEDULED JOBS *****
//SELECT    EXEC PGM=CA7INTF
//STEPLIB   DD DISP=SHR,DSN=ASG.SJJSLMOD           <== ASG LOAD LIB
//I1FILE    DD DSN=*.SCHEDULE.SYSPRINT,DISP=(OLD,DELETE)
//O1FILE    DD DSN=ASG.SELECT.CARDS,              <== OUTPUT DSNAME
//          DISP=(,CATLG,DELETE),SPACE=(TRK,(10,2),RLSE),
//          UNIT=SYSDA,                            <== VALID UNIT
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//SYSUDUMP  DD SYSOUT=*
//*
//***** EXECUTE JOB/SCAN *****
//JOBSCAN   EXEC JS6PROC,PRM=S
//JLJOB     DD DSN=YOUR.JCLLIB,DISP=OLD           <== INPUT JCL LIBRARY
//SYSIN     DD DSN=*.SELECT.O1FILE,DISP=(OLD,DELETE)
```

## CA-Scheduler Interface

To validate a set of jobs accurately, it is sometimes desirable to process the jobs in run-time order. ASG provides a program and JCL to interface JOB/SCAN with the CA Scheduler.

### The Interface

The interface consists of a two-step job that:

- Executes the ASG CASINTF program to create a list of jobs extracted from a CA Scheduler forecasting report.
- Executes JOB/SCAN to validate the jobs in run-time order.

## Establishing the Interface: Procedure Overview

### To establish the interface

1. Verify that the interface program shipped by ASG is correct for your site's release of CA Scheduler.
  - a. Print a source listing of the CASINTF COBOL program in the SJJSSAMP library.
  - b. Check the input File Description and the input record layout, and modify the COBOL code if necessary.
  - c. Compile and link the program. (To compile JOB/SCAN COBOL exits, use the ISPF edit macro CCEXIT in SJJSSAMP.)
  - d. Verify that the program is creating a series of SELECT= statements, which will be used by JOB/SCAN.
2. Produce the Schedules/Jobs Selected CA Scheduler forecasting report and save it as a disk dataset with these attributes: LRECL=133, RECFM=FBA. (See the CA Scheduler product documentation for details.)
3. Modify the Interface JCL and Submit the Job
  - a. Modify the JCL contained in the CASCHED member in the SJJSSAMP library. Make sure the INFILE DD statement points to the dataset containing the CA Scheduler report produced earlier. Modify the rest of the JCL according to the directions contained in the member.
  - b. Submit the job.

This shows sample JCL for the CA Scheduler interface (in the CASCHED member of the SJJSSAMP library):

### Sample CASCHED JCL

```
//CASCHED JOB (ACCT), 'PROGRAMMER NAME'
//*****
//* INTERFACE TO CA-SCHEDULER
//* REF: SJJSSAMP(CASCHED)
//*
//* NOTE: MODIFY JOB STATEMENT AND ITEMS INDICATED BY ARROWS <==
//*****
//***** GENERATE SELECTION STATEMENTS FOR THE SCHEDULED JOBS *****
//SELECT EXEC PGM=CASINTF
//STEPLIB DD DISP=SHR,DSN=ASG.SJJSIMOD <== ASG LOAD LIB
//INFILE DD DISP=OLD,DSN=CASCHED.REPORT.FILE <== CA SCHED FILE
//OUTFILE DD DSN=DSSI.SELECT.CARDS, <== OUTPUT DSNNAME
// DISP=(,CATLG,DELETE),UNIT=SYSDA, <== VALID UNIT
// SPACE=(TRK,(20,5),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//SYSUDUMP DD SYSOUT=*
//*
//***** EXECUTE JOB/SCAN *****
//JOBSCAN EXEC JS6PROC,PRM=S
//JLJOB DD DSN=YOUR.JCLLIB,DISP=OLD <== INPUT JCL LIBRARY
//SYSIN DD DSN=*.SELECT.OUTFILE,DISP=(OLD,DELETE)
```

## CA-Driver Interface

CA-Driver can be accessed by the CA-7 Scheduler to provide additional job tailoring. If both CA-7 and CA-Driver are active, then use the JOB/SCAN CA-Driver interface to simulate the functions both CA-7 and CA-Driver perform at JOB submission time.

### The Interface

The JOB/SCAN CA-Driver interface consists of a three-step job (CADRVJS, see figure) that:

- Executes the CA-7 LJCK program to extract the JCL from CA-7 for the job specified. CA-7 accesses CA-Driver and resolves the variables.
- Executes the ASG CADRIVER program to process the output list from LJCK and produce the JCL just as it would be submitted to the JES Reader.
- Executes JOB/SCAN to analyze the JCL.

### Establishing the Interface

#### *To establish the CA-Driver interface*

1. Review the CADRIVER REXX program in the SJJSSAMP library. The program should work as-is but it can be modified as necessary.
2. Copy the CADRIVER program to the appropriate library for execution.

### Using the Interface

#### *To use the CA-Driver interface*

(These steps correspond to numerical labels 1-4 in figure.)

1. Determine the requirements for accessing CA-7 in batch mode. Insert this information at label 1.
2. Insert the name of the JOB at label 2.
3. Insert the location where the program CADRIVER is loaded at label 3.
4. Insert the usual JOB/SCAN selection criteria at label 4.
5. Submit the job.
6. JOB/SCAN produces an SJL; review this report and perform any indicated actions.

Future executions of CADRVJS require only that the JOB name be changed (see note 2 in figure).

## CADRVJS sample JCL

```

//CADRVJS JOB (ACCT), 'ASG - CADRIVER',MSGCLASS=X,
//          NOTIFY=&SYSUID
//*****
//* CA7DRVJS- INTERFACE TO CA-DRIVER
//*****
//*
//*****
//***** GENERATE CA-DRIVER REPORT *****
//*****
//STEP1 EXEC PGM=SASSBSTR
1. //CA7CMDS DD DSN=YOUR.CA7.CMDS,DISP=SHR <== CA-7 CMD FILE
1. //BATCHIN DD DSN=YOUR.CA7.BATCH,DISP=SHR <== CA-7 INPUT
1. //BATCHOUT DD DSN=YOUR.CA7.BATCHOUT,DISP=SHR <== CA-7 OUTPUT
//SYSPRINT DD DSN=&&CADRVR,DISP=(,PASS),
//          SPACE=(TRK,(30,10),RLSE),UNIT=SYSDA, <== VALID UNIT
//          DCB=(RECFM=FBA,LRECL=133,BLKSIZE=6118)
//SYSIN DD DATA,DLM=ZZ
/LOGON PASSWORD
2. LJCK,JOB=YOURJOB,LIST=ONLY <===== CHANGE TO YOUR JOB NAME
/LOGOFF
ZZ
//SYSUDUMP DD SYSOUT=*
//*****
//* EXTRACT THE JCL FROM THE REPORT *****
//*****
//ISPF EXEC PGM=IKJEFT01,DYNAMNBR=25,REGION=1024K
//ISPLOG DD SYSOUT=*,
//          DCB=(RECFM=VA,LRECL=125,BLKSIZE=129)
//*
3. //SYSEXEC DD DISP=SHR,DSN=YOUR.EXEC <== CHANGE
3. //SYSPROC DD DISP=SHR,DSN=YOUR.EXEC <== CHANGE
//*
//INPUT DD DSN=*.STEP1.SYSPRINT,DISP=(OLD,DELETE)
//OUTPUT DD DSN=&&JCLOUT,DCB=(LRECL=80,BLKSIZE=0),
//          UNIT=SYSDA,SPACE=(TRK,(5,1)),DISP=(,PASS)
//JPDIAG DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSTSIN DD *
CADRIVER
//*****
//* EXECUTE JOB/SCAN *****
//*****
//JSSTEP1 EXEC JS6PROC,PRM=S
//SYSIN DD *
INPUT SELECTION
4. DDNAME=INPUT <== INPUT JCL LIBRARY
4. SOURCE TYPE=SEQUENTIAL <== TYPE OF INPUT LIB
REPORT OPTIONS
4. OPTIONS IN EFFECT =PRINT <== PRINT OPE REPORT?
4. STRUCTURED JCL LISTING=PRINT <== PRINT SJL REPORT?
//INPUT DD DSN=*.ISPF.OUTPUT,DISP=(OLD,DELETE)
//*

```

## CONTROL-M Scheduler Interface

Three options are available for interfacing JOB/SCAN with the CONTROL-M Scheduler:

- A dynamic CONTROL-M interface to resolve the %% variables. See [Resolving %% Variables with the CONTROL-M Scheduler Interface](#) for information on using this method.

- A batch interface to CONTROL-M to validate production jobs in run-time order. See [Validating Jobs in Run-time Order with the CONTROL-M Scheduler Interface](#) for information using this method.
- An interface to validate jobs when variable information is stored in the job scheduling definition. See [Validating Jobs in the Job Scheduling Definition with the CONTROL-M Scheduler Interface](#) for information using this method.



With all of these options, unless your CONTROL-M load library is in your LNKLST concatenation, add it to the JOBLIB or STEPLIB for JOB/SCAN execution.

### Resolving %% Variables with the CONTROL-M Scheduler Interface

Production JCL run by CONTROL-M often contains CONTROL-M and user %% variables and %%CALCDATE functions. The %% variables need to be resolved before the jobs can be validated. JOB/SCAN's dynamic CONTROL-M interface invokes special CONTROL-M routines to resolve %% variables in the JCL. This interface can be used during foreground and batch executions of JOB/SCAN.

### Establishing the Interface

#### To establish the interface

##### 1. Identify Dataset Names

Identify the names of the CONTROL-M files:

- DAGLOBAL: File containing the Auto-Edit variables.
- PRTDBG: Internal CONTROL-M debugging (may be allocated as DUMMY).
- DACAL: Optional IOA Calendar Facility library.
- DAPARM: Required for CONTROL-M, Release 6.0

##### 2. Allocate the files in one of these ways:

- For batch processing, add a DD statement to JS6PROC; for foreground processing, add the DD statement to the logon PROC. For example:

```
//DAGLOBAL DD DSN=PROD.AUTO.EDIT,DISP=SHR
//DAPARM   DD DISP=SHR,DSN=NDS.IOA602.PARM
//        DD DISP=SHR,DSN=NDS.IOA602.IOAENV
//PRTDBG   DD DUMMY
//DACAL    DD DISP=SHR,DSN=calendar.library
```

If you want JOB/SCAN to load CTL-M exits from JLLOD, add this allocation statement:

```
//JLLOD DD DISP=SHR,DSN=JOB.SCAN.SJJSIMOD
```

- Allocate the files in a CLIST for foreground execution. For example:

```
ALLOC FI (DAGLOBAL) DA ('PROD.AUTO.EDIT') SHR REU
ALLOC FI (PRTDBG)   DUMMY
ALLOC FI (DACAL)    DA ('calendar.library') SHR REU
ALLOC FI (DAPARM)   DA ('NDS.IOA602.PARM' 'NDS.IOA602.IOAENV') SHR
```

- Add statements to the File Allocation section of the appropriate SJJSPARM library member for both foreground and batch execution. For example:

```
DAGLOBAL    =DSN=PROD.AUTO.EDIT, SHR
PRTDBG      =DUMMY
DACAL       =DSN=calendar.library, SHR
DAPARM      =DSN=NDS.IOA602.PARM, SHR
            =DSN=NDS.IOA602.IOAENV, SHR
```

If you want JOB/SCAN to load CTL-M exits from JLLOD, add this allocation statement to the CLIST for the environment:

```
ALLOC F (JLLOD) DA (JOB.SCAN.SJJSLSMOD) REU SHR
```

To load CTL-M exits from the JOB/SCAN load library, add SJJSLSMOD to your ISPLLIB concatenation:

```
//JLLOD DD DISP=SHR, DSN=JOB.SCAN.SJJSLSMOD
```

### 3. Turn On the CONTROL-M Interface.

Add the CONTROL-M parameter to the Scheduler (SCHED) Run Option:

```
SCHEDULER =CONTROL-M
```

CONTROL-M routines are now automatically invoked each time JOB/SCAN is run on a job containing %% variables.

## Validating Jobs in Run-time Order with the CONTROL-M Scheduler Interface

To validate a set of jobs accurately, it is sometimes desirable to process the jobs in run-time order. ASG provides a program and JCL to interface JOB/SCAN with the CONTROL-M Scheduler for validating all jobs on the schedule for a given time period. The interface extracts JCL from the CONTROL-M environment and writes it to a sequential file; the file is then input to JOB/SCAN for processing.

### The Interface

The interface consists of a two-step job that:

- Executes the CONTROL-M CTMTAPUL PROC to extract a set of jobs and place them in a sequential file.
- Executes JOB/SCAN to validate the jobs in their scheduled order.

## Establishing the Interface

### To establish the interface

1. Modify the interface JCL according to the directions contained in the CONTROLM member of the SJJSAMP library.
2. Run the CONTROL-M Simulation Facility using appropriate Active Jobs File and Conditions/Resources File as input. Simulation Facility output will be a new Log File that is forecast for the period simulated. This will be input into the CTMTAPUL utility.
3. Submit the job.



The figure shows sample JCL for the CONTROL-M Scheduler interface (in the CONTROLM member of the SJSSAMP library):

#### Sample CONTROLM JCL

```
//CONTROLM JOB (ACCT), 'PROGRAMMER NAME'
//*****
//*   INTERFACE TO THE CONTROL-M SCHEDULER (TO PROCESS JOBS IN RUN-TIME
//*   ORDER)
//*   REF:   SJSSAMP (CONTROLM)
//*
//*   NOTE:  MODIFY JOB STATEMENT AND ITEMS INDICATED BY ARROWS <==
//*****
//*
//***** GENERATE A FILE OF JOBS SCHEDULED BY CONTROL-M *****
//CTM1      EXEC CTMTAPUL                               <== CNTL-M EXEC PROC
//TAPEPULL.TAPULIN DD *
JCLFILE ONLY
//TAPEPULL.DAJCLOUT DD DSN=&&JCLFILE,
//              DISP=(,PASS),UNIT=SYSDA,              <== VALID UNIT
//              SPACE=(TRK,(30,30)),
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=6320)
//*
//***** EXECUTE JOB/SCAN *****
//JOBSCAN   EXEC JS6PROC,PRM=S
//SYSIN     DD *
INPUT SELECTION
SOURCE TYPE=SEQUENTIAL
//JLJOB     DD DSN=&&JCLFILE,
//              DISP=(OLD,DELETE)
```



Additional code from New Dimensions (CONTROL-M) may be required to support JOB/SCAN as an exception to the CTSMS02 security exit.

#### Validating Jobs in the Job Scheduling Definition with the CONTROL-M Scheduler Interface

JOB/SCAN provides sample JCL that validates a set of jobs defined in a scheduling library table (SJSSAMP member CTLMAE). This sample JCL can be used to resolve variables defined in the scheduling library.

#### The Interface

The interface consists of a two-step job that:

- Executes the CONTROL-M batch utility CTMAESIM to resolve all CONTROL-M variables and statements and place them in a sequential file using the SCHEDLIB definitions.
- Executes JOB/SCAN to validate the jobs (from the sequential file).

#### Establishing the Interface

##### *To establish the interface*

1. Modify the interface JCL according to the directions contained in the CTLMAE member of the SJSSAMP library.
2. Submit the job.

Member CTLMAE (see the below figure), in the JOB/SCAN SJSSAMP, contains sample JCL to invoke the CONTROL-M Batch Utility Program (CTMAESIM). See CONTROL-M Users Guide for information concerning CTMAESIM JCL or control parms.



- JCLLIB concatenates the JOB/SCAN and CONTROL-M procedure libraries.
- DASUBMIT is The output dataset for the batch CONTROL-M simulator to be passed to the JOBSCAN run. JIJOB and DASUBMIT must be the same name.
- SCHEDLIB points to the library containing the CONTROL-M Scheduling Definitions.
- TABLE contains the job scheduling definition.
- Job\* is the Job scheduling pattern mask.
- JSCANPLB – Library in which JOB/SCAN procedure is located.
- DAPARM - this DD statement is required for CONTROL-M 6.0.

## SJSSAMP member CTLMAE

```

***** Top of Data *****
//CTLMAE JOB (ACCT),'PROGRAMMER NAME' <=====
//*****
//* REF: SJSSAMP (CTMAESIM)
//*
//* THIS JOB RUNS THE CONTROL-M BATCH UTILITY PROGRAM CTMAESIM AND
//* PASSES ITS INPUT (WITH ALL JOBS AUTO-EDIT VARIABLES RESOLVED)
//* TO JOBS CAN FOR VALIDATION.
//*
//*
//* NOTE: MODIFY ALL STATEMENTS INDICATED BY AN ARROW <=====
//*
//* NOTE: REFER TO THE CONTROL-M USER MANUAL FOR DETAILS ON CTMAESIM
//*****
// SET HLQ='---HLQ---' <=====
// SET CNTLMPCRC='---CONTROL.M.PROCLIB---' <=====
// SET DAGLOBAL='---CNTLM.GLOBAL.LIB---' <=====
// SET JSCANPLB='---JOBS CAN.PROCLIB---' <=====
//*
// JCLLIB ORDER=(&HLQ..SJSINST,
// &CNTLMPCRC,&JSCANPLB)
//*THE FOLLOWING STEP EXECUTES THE CONTROL-M SIMULATOR. ALL CONTROL-M
//* AUTO-EDIT VARIABLES AND STATEMENTS WILL BE RESOLVED.
//CTMAESIM EXEC CTMAESIM
//DAGLOBAL DD DSN=&DAGLOBAL,DISP=SHR
//DAPARM DD DISP=SHR,DSN=NDS.IOA602.PARM
// DD DISP=SHR,DSN=NDS.IOA602.IOAENV
//DASUBMIT DD DISP=(,PASS),DSN=&&JCLFILE,UNIT=SYSDA, <===== VALID UNIT
// SPACE=(CYL,(1,1)),DCB=(LRECL=80,BLKSIZE=0,RECFM=FB)
//SYSPRINT DD SYSOUT=*
//* VARIABLES DEFINED IN THE SCHEDULING LIBRARY CAN BE USED
//DASIM DD *
// ODATE ?YOUR DATE? (Original scheduling date of job) <=====
// SCHEDLIB ?YOUR.SCHEDULING.LIBRARY? <=====
// TABLE ?YOURTABLE? <=====
// JOB * <=====
// SUBMIT
//*
//***** EXECUTE JOB/SCAN *****
//JOBS CAN EXEC JS6PROC,PRM=S
//SYSIN DD *
INPUT SELECTION
SOURCE TYPE=SEQUENTIAL
//JLJOB DD DSN=&&JCLFILE,
// DISP=(OLD,DELETE)
***** Bottom of Data *****

```

## DCMS Scheduler Interface

To validate a set of jobs accurately, it is sometimes desirable to process the jobs in run-time order. ASG provides a program and JCL to interface JOB/SCAN with the DCMS Scheduler.

### The Interface

The interface consists of a two-step job that:

- Executes the ASG DCMSINTF program to create a list of jobs extracted from the DCMS Scheduler.
- Executes JOB/SCAN to validate the jobs in run-time order.

## Establishing the Interface

### To establish the interface

1. Verify the Interface Program
  - a. Print a source listing of the DCMSINTF COBOL program in the SJJSSAMP library.
  - b. Check the input File Description and the input record layout, and modify the COBOL code if necessary.
  - c. Compile and link the program. (To compile JOB/SCAN COBOL exits, use the ISPF edit macro CCEXIT in SJJSSAMP.)
  - d. Verify that the program is creating a series of SELECT= statements, which will be used by JOB/SCAN.
2. Modify the Interface JCL and Submit the Job
  - a. Modify the JCL contained in the DCMSCHED member in the SJJSSAMP library. Make sure the SCHEDULE DD statement points to the DCMS file containing the job names for the scheduled run. (This is the file that updates the DCMS.JET file each time a schedule is set up.)
  - b. Modify the rest of the JCL according to the directions contained in the member.
  - c. Submit the job.

The figure shows sample JCL for the DCMS Scheduler interface (in the DCMSCHED member of the SJJSSAMP library):

#### Sample DCMSCHED JCL

```
//DCMSCHED JOB (ACCT), 'PROGRAMMER NAME'
//*****
//* INTERFACE TO DCMS SCHEDULER
//* REF: SJJSSAMP(DCMSCHED)
//*
//* NOTE: MODIFY JOB STATEMENT AND ITEMS INDICATED BY ARROWS <==
//*****
//***** GENERATE SELECTION STATEMENTS FOR THE SCHEDULED JOBS *****
//SELECT EXEC PGM=DCMSINTF
//STEPLIB DD DSN=ASG.SJJSIMOD,DISP=SHR <== ASG LOAD LIB
//SCHEDULE DD DSN=DCMS.TAPE,DISP=OLD <== DCMS DATA SET
//DCMSINTF DD DSN=ASG.SELECT.CARDS, <== OUTPUT DSNAME
// DISP=(,CATLG,DELETE),UNIT=SYSDA, <== VALID UNIT
// SPACE=(TRK,(10,5),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//SYSUDUMP DD SYSOUT=*
//*
//***** EXECUTE JOB/SCAN *****
//JOBSCAN EXEC JS6PROC,PRM=S (TO EXECUTE JOB/SCAN)
//JLJOB DD DSN=YOUR.JCLLIB,DISP=OLD <== INPUT JCL LIBRARY
//SYSIN DD DSN=*.SELECT.DCMSINTF,DISP=(OLD,DELETE)
```

## ESP Scheduler Interface

Technical documentation for JOB/SCAN's ESP Scheduler interface is available in [User Information - JOB/SCAN with ESP](#) documentation. It contains complete details about the installation and use of the ESP support.

## JOBTRAC Scheduler Interface

ASG provides a JOBTRAC interface simulating some of these functions of JOBTRAC:

- Automatic JCL Override
- JCL Management Facility
- Automatic Date Insertion

### Resolving Variables

The JOBTRAC interface is invoked as a selection exit for JOB/SCAN using the JTRC2XIT member found in the SJSSAMP library.

See [Selection Exit Interface](#) for general information about selection exits.

### Automatic JCL Override

Automatic JCL Override allows the user to include or exclude specific JCL lines in a job.

### JCL Management Facility

The JCL Management Facility uses variables located in another dataset. The JOBTRAC Selection Exit assumes the jobs being scanned use location 000 unless the LOCATION= parameter is coded. If the parameter is coded, variables specified for that location are loaded and substitution is performed on that basis. See [Selection Exit Parameters](#) for more information.

### Automatic Date Insertion

Automatic Date Insertion uses JOBTRAC variables for date manipulation. Workdays include all weekdays, and exclude Saturday and Sunday. Holidays are not recognized. See [Selection Exit Parameters](#) for more information.

### Selection Exit Parameters

There are five parameters used to control JOBTRAC variables:

Parameter	Description	Examples
LOC=LOCATION=	Defines the location for this job; valid numbers are three-digits from 001 to 254.	LOCATION=002
OSD=OSDDATE=	Sets the original scheduled date; must be specified as Julian date.	OSDDATE=94103
TOD=TODDATE=	Sets the current processor date; must be specified as Julian date.	TODDATE=94103
UDEF=	One or more user-defined process periods that are accepted as true and action taken for current run.	UDEF=1A(UDEF1 and UDEFA are accepted as true).
SCL=	Names a member of JOBTRAC-SCL-DSN.	SCL=member

Parameters may be specified in any order, separated by commas. If not specified, OSDDATE and TODDATE default to the current date. Overriding one of these dates has no effect on the other.

DEF= and SCL= are mutually exclusive.

### **To establish the interface**

1. Determine which version of the exit you need to use.
2. Locate the exit in SJJSSAMP.
3. Change JOBTRAC-VARIABLE-DSN in the exit program (working-storage section) to the JOBTRAC symbolic PDS name (JMF symbolic files and JOBTRAC production date card member files).
4. Change JOBTRAC-SCL-DSN in the exit program to the JOBTRAC schedule PDS name (contains the automatic date insertion rules). This is only necessary if SCL= is specified for the PARM.
5. Assemble and link-edit the READDSN program (an Assembler language program (also in SJJSSAMP) that allocates and reads files).
6. After reviewing the information in the topic - [Setting Up Optional Interfaces and Exits](#), compile the exit using the CCEXIT edit macro found in SJJSSAMP.

As an alternative, the JOB/SCAN Standards Prompter can be used for the compilation. For more information about the Standards Prompter, see [Select Lists Detail Statement](#) and [Standards Information](#). Allocate DDNAME JTRCDATE specifying the JOBTRAC name dataset containing member PRODDATE. For example:

```
//JTRCDATE DD DISP=SHR,DSN=JOBTRAC.DATEFILE(PRODDATE)
```

This can be done in a logon PROC or JOB/SCAN Startup CLIST. If the allocation is not done, the exit allocates the dataset based on the name supplied in Step 3. This will not be deallocated.



To use the international date form (mm/dd/yy), see the remarks in the exit program.

### **JOBTRAC Scheduler Interface: Validating Jobs in Run-time Order**

To validate a set of jobs accurately, it is sometimes desirable to process the jobs in run-time order. ASG provides a program and JCL to interface JOB/SCAN with the JOBTRAC Scheduler. Use the instructions in this section with JOBTRAC releases prior to Release 3.2. For JOBTRAC Release 3.2 and later, use the instructions found in the section, [JOBTRAC Scheduler Interface: Validating Jobs with JOBTRAC Release 3.2 and later](#).

The interface consists of a three-step job that:

- Executes a JOBTRAC utility to produce a list of jobs in run-time order.
- Executes the ASG JTSELECT program to format the job names into a list of jobs to be validated by JOB/SCAN.
- Executes JOB/SCAN to validate the jobs in run-time order.

**To establish the interface**

1. Verify the Interface Program
  - a. Print a source listing of the JTSELECT COBOL program in the SJJSSAMP library.
  - b. Check the input File Description and the input record layout, and modify the COBOL code if necessary.
  - c. Compile and link the program. (To compile JOB/SCAN COBOL exits, use the ISPF edit macro CCEXIT in SJJSSAMP.)
  - d. Verify that the program is creating a series of appropriate SELECT= statements, which will be used by JOB/SCAN.
2. Modify the Interface JCL and Submit the Job
  - a. Determine the criteria for extracting job names in execution sequence from the JOBTRAC scheduler.
  - b. Add appropriate JOBTRAC utility JCL to the first step of the JCL contained in the JTRACSCH member of the SJJSSAMP library. (See the JOBTRAC product documentation for information on producing the LIST SCHEDULE report.)
  - c. Modify the rest of the JTRACSCH JCL according to the directions contained in the member.
  - d. Submit the job.

**JOBTRAC Scheduler Interface: Validating Jobs with JOBTRAC Release 3.2 and later**

To validate a set of jobs accurately, it is necessary to process the jobs in run-time order. ASG provides a facility to interface JOB/SCAN with the JOBTRAC Scheduler. Use the instructions for this interface if you're running JOBTRAC Release 3.2 or greater. For JOBTRAC prior to Release 3.2, use the instructions found in the previous section, [JOBTRAC Scheduler Interface: Validating Jobs with JOBTRAC Release 3.2 and later](#).

The interface consists of a two-step job that:

- Executes a CA-JOBTRAC utility to produce a list of jobs in run-time order.
- Executes JOB/SCAN to validate the jobs in run-time order.

**To establish the interface**

1. Verify the Interface Program.
  - a. Run JREPORT3 DISPLAY SCHEDULE - sample JCL is located in SJJSSAMP member JTSCHED.
  - b. Check the output for a heading of JOBTRAC REPORTER Vn.n DISPLAY SCHEDULE LISTING. The displayed release must be V3.2 or greater.
2. Modify the JOBTRAC Utility JCL.
  - a. Determine the criteria for extracting job names in execution sequence from the CA-JOBTRAC scheduler.
  - b. Add JREPORT3 statements to limit the jobs that will be included in the validation run. Refer to the *CAJOBTRAC Reports, Maintenance, and Recovery Guide, Release*

3. 4 *MVS*. This guide contains information on the END, OSD, and TARGET keywords (used to limit the time frame), and the JOB, LOCID, and O keywords (used to limit the jobs to be included in the validation).

3. Allocate the files in one of two ways:

- Add the DD statements to the JOB/SCAN job JCL. For example:

```
//FORECAST DD DISP=SHR, DSN=JREPORT3.DISPLAY.SCHEDULE
//JLJOB    DD DISP=SHR, DSN=PDS1.JOBS
//         DD DISP=SHR, DSN=PDS2.JOBS
//         DD DISP=SHR, DSN=PDS3.JOBS
```

**Or**

- Add statements to the File Allocations section of the appropriate SJJSPARM member. For example:

```
FORECAST =SHR, DSN=JREPORT3.DISPLAY.SCHEDULE
JLJOB    =SHR, DSN=PDS1.JOBS
         =SHR, DSN=PDS2.JOBS
         =SHR, DSN=PDS3.JOBS
```

4. Add the JOBTRAC parameter and the optional Selection Exit to the SCHEDULER Run Option in the appropriate SJJSPARM member.
5. Submit the job (see member JTSCHEM in SJJSSAMP library).

## TWS z/OS Scheduler Interface

The JOB/SCAN interface for TWS z/OS Scheduler is fully explained in [User Information - JOB/SCAN with TWS](#) documentation. This documentation describes the full range of JOB/SCAN facilities for integrating JOB/SCAN into your TWS workflow.

## ASG-Zeke Scheduler Interface: \$ Symbolic Parameters

The ASG-Zeke (herein called ZEKE) interface substitutes appropriate values into JCL containing ZEKE symbolic parameters. ZEKE parameters are coded as a name preceded by one or two dollar signs (\$). The ZEKE interface prevents JOB/SCAN from issuing error messages when it encounters these parameters in JCL that it is processing. The ZEKE interface also ensures that all JCL elements (for example, dataset names) are fully resolved.

The ZEKE interface is invoked as a selection exit for JOB/SCAN.

### Prepare the Interface

#### *To establish the interface*

1. Compile the ZEKEVARS source program in the SJJSSAMP library using the edit macro CCEXIT.
2. Make the ZEKE load library available to the job or TSO session via STEPLIB, link list, etc.

### Activate the Interface

Activate the interface as follows:



- For panel processing: enter ZEKEVARS as the selection exit name on the JCL Validation panel:

```
SELECTION EXIT ==> ZEKEVARS      Parm  ==>>>
```

- For Edit macro processing: modify the input selection section of JSOPTM in the SJJSPARM library by inserting this line:

```
SELECTION EXIT      =ZEKEVARS
```

- For batch processing: insert this line in the INPUT SELECTION section:

```
SELECTION EXIT      =ZEKEVARS
```

ZEKE 4.2 introduced the capability to run multiple ZEKE subsystems under a single MVS. If you are on ZEKE 4.2 or later, and have multiple ZEKE subsystems (such as a test and a production ZEKE) running on the same machine, you may wish to specify which ZEKE should resolve the variables. ZEKEVARS accepts an optional parameter, as follows:

- For panel processing: enter ZEKEVARS as the selection exit name on the JCL Validation panel. Specify *SUBSYS=subsysname* as the parm:

```
SELECTION EXIT ==> ZEKEVARS Parm  ==> SUBSYS=SSSI
```

- For Edit macro processing: modify the input selection section of JSOPTM in the SJJSPARM library by inserting this line:

```
SELECTION EXIT =ZEKEVARS, 'SUBSYS=SSSI' <==
```

- For batch processing without panels: insert this line in the input selection section:

```
SELECTION EXIT =ZEKEVARS, 'SUBSYS=SSSI'
```

If no subsystem is specified, ZEKEVARS uses the ZEKE installation default (SSSI).



- ZEKEVARS must have the appropriate ZEKE security access to function properly. ZEKE has two types of security available:
  - Internal security is performed entirely by ZEKE and is administered through ZEKE panels.
  - External security is an interface to RACF, Top Secret or ACF2.
- ZEKE, as supplied on its installation tape, has:
  - internal security activated and external security deactivated.
  - The USERID OPERATOR defined and it has write access to everything.
    - If OPERATOR authority has not been removed or altered, ZEKEVARS will work as supplied.
    - If OPERATOR has been removed or its authority reduced, another suitable USERID must be provided. The USERID must have read (R) access to the ZEKE function Variable. Write (W) access is usable but not required, because JOB/SCAN does not update any datasets in ZEKE.

## Results

The original JCL statement containing the ZEKE symbolic parameter(s) is converted into a `/**` comment followed by a fully resolved JCL statement.

Under certain conditions such as the inclusion of ZEKE variables within SYSIN statements, this may not be desirable. The generated comment statements may be omitted by specifying NOCOMM in the selection exit PARM field.

## Example

Assume that ZEKE variable `$OUT1` has been set to a value of `A`, `$FILE` has been set to `PAYROLL`, `$NEW` has been set to `BACKUP` and `$ALL` has been set to `CYL`.

Input JCL:

```
//ZEKEJOB JOB (ACCT),'programmer-name',
//          CLASS=A,MSGCLASS=X
//STEP1   EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=$OUT1
//SYSIN   DD DUMMY
//SYSUT1  DD DISP=OLD,DSN=A.B.$FILE
//SYSUT2  DD DISP=(,CATLG),DSN=A.B.$NEW,
//          UNIT=DISK,SPACE=( $ALL,(5,1) ),
//          DCB=(RECFM=FB,BLKSIZE=3990)
```

Output JCL:

```
//ZEKEJOB JOB (ACCT),'programmer-name',
//          CLASS=A,MSGCLASS=X
//STEP1   EXEC PGM=IEBGENER
//*SYSPRINT DD SYSOUT=$OUT1
//SYSPRINT DD SYSOUT=A
//SYSIN   DD DUMMY
//*SYSUT1  DD DISP=OLD,DSN=A.B.$FILE
//SYSUT1  DD DISP=OLD,DSN=A.B.PAYROLL
//*SYSUT2  DD DISP=(,CATLG),DSN=A.B.$NEW,
//SYSUT2  DD DISP=(,CATLG),DSN=A.B.BACKUP,
//*          UNIT=DISK,SPACE=( $ALL,(5,1) ),
//          UNIT=DISK,SPACE=(CYL,(5,1) ),
//          DCB=(RECFM=FB,BLKSIZE=3990)
```

## Validation at Job Submission Time

ASG provides a ZEKE interface which makes successful JOB/SCAN Validation a prerequisite to ZEKE submitting a job to JES. The interface is written as a ZEKE14E User Exit and supports complete JCL Validation, including compliance with in-house standards.

### *To establish the interface*

1. Assemble the Exit.

Assemble the SJJSSAMP(ZEKE14E) member using the guidelines specified in the ZEKE documentation. Make sure the exit is linked with any special attributes required by the installed release of ZEKE, such as serial reusability (REUS).

2. Adding Required DD Statements to ZEKE.

SJJSSAMP member ZEKEDOC provides examples of DD statements that must be added to allow JOB/SCAN to run. Both JCL and TSO allocate examples are provided.



The JOB/SCAN exit only inspects the JCL, so do not set the \$14DFLAG to E. JOB/SCAN exit will return condition code of 4 if it detects any problems with the JCL. This signals ZEKE to terminate processing and place the event on hold. Otherwise, it will return a condition code of 0 to signal ZEKE to continue processing.

### Validation of Jobs in Run-time Order

ASG provides a ZEKE interface which can process the jobs in run-time order. This interface consists of a three step job that:

- Executes an ZEKE utility program to produce a list of jobs in run-time order.
- Executes the ZEKESCH program to format the list of jobs to be validated by JOB/SCAN.
- Executes JOB/SCAN to validate the jobs in run-time order.

#### *To establish the interface*

1. Modify the Interface JCL, SJJSSAMP member ZEKESCH
  - a. Determine the criteria for extracting job names in the execution sequence from the ZEKE scheduler.
  - b. Add ZEKEUTL statements to control the jobs included in the validation. Refer to the Report Writer information contained in the ZEKE documentation for more details.
  - c. Add a ZEKESCH control statement if necessary (optional SYSIN dd statement).

**JOB**COL= nn

nn is the report column number in which the jobname starts. Default - 2.

**LASTEVENT**COL=nn

nn is the column number in which the event number ends. Default - 35.

**SELECTION EXIT**=*exitname*

*exitname* is the name of a selection exit specification to pass to JOB/SCAN.

Note that **JOB**COL= and **LASTEVENT**COL= each start in position 1 of the control statement and **SELECTION EXIT** = starts in position 2 or greater.

- a. Modify the ZEKESCH JCL to conform to site standards.
2. Submit the job.

## BETA42 Scheduler Interface

To validate a set of jobs accurately, it is necessary to process the jobs in run-time order and resolve scheduler variables. ASG provides a program and the JCL to interface JOB/SCAN with the BETA42 Scheduler.

### The Interface

The interface consists of a job that:

- Executes a BETA42 utility to produce a list of jobs in run-time order.
- Executes the ASG J00YBT42 program to retrieve the JCL and resolve variables.
- Executes JOB/SCAN to validate the jobs in run-time order.

## Establishing the Interface

### *To establish the interface*

1. Verify that your site has BETA42 Version 1 Release 9 (with fix SS195109) or later.
2. Determine the criteria for extracting job names in execution sequence from the BETA42 scheduler.
3. Modify the SCHR0040 BETA42 utility control statements found in the first step of the BETA42 member of the SJJSSAMP library.
4. Modify the rest of the SJJSSAMP member BETA42 according to the directions contained in the member.
5. Submit the job.

# Security Interfaces

JOB/SCAN's interfaces to security products makes sure that jobs are validated with the proper security authorizations in place to give the best run-time simulation possible.

- [Security Interface Overview](#)
- [Security Interface Functions](#)
- [Security Interface Description](#)

## Security Interface Overview

The security interface of JOB/SCAN works with security software such as RACF, CA-TOP SECRET, and CA-ACF2/MVS to provide these functions:

- Dataset Name Authorization Checking

JOB/SCAN validates to see if the userid running JOB/SCAN has proper authorization to read, write, create, and delete datasets in the JCL.

This function requires the DSNAME parameter. (See [Establishing the Interface](#))

- Alternate Userid Validation

If the JOB statement contains a USER= parameter or a //\*LOGONID statement exists, JOB/SCAN uses that userid instead of the TSO userid executing JOB/SCAN to validate dataset access.



The use of the USERID parameter from the JCL is only available when the JOB/SCAN program being executed has access to the library that it will validate. If JOB/SCAN doesn't have access to a library, the library will not be opened even if the USERID on the job card has access to the library.

To perform this function, MVS requires JOB/SCAN to be APF-authorized. If this function does not need to be performed each time JOB/SCAN executes, foreground performance can be improved by specifying *N* for the Security Exit DSNAME Check in the JOB/SCAN Processing Options panel.

This function also requires the DSNAME parameter. (See [Establishing the Interface](#))

- Establishing the Interface

The security interface is invoked with this Run Option:

```
SECURITY EXIT=name, DSNAME
```

For example:

```
SECURITY EXIT=SAF2, DSNAME
```

Use the SHOW RUN command from any JOB/SCAN panel to display the status of the security interface.

## Security Interface Functions

The security interface is a user exit routine controlled by the Security Exit Run Option; it interfaces JOB/SCAN with the RACF, CA-ACF2, or CA-TOP SECRET security systems.

This interface consists of three functions: JOB/SCAN access control, DSNNAME validation, and JOB/SCAN termination. This interface does not interfere with normal security logging procedures.

### JOB/SCAN Access Control

The JOB/SCAN Access Control function interfaces with a security package (CA-ACF2/MVS, CA-TOP SECRET, or RACF) to verify that the JOB/SCAN job or TSO session has the proper access privileges to files prior to the actual access of those files. This prevents JOB/SCAN from causing security violations and possibly an S913 Abend.

In analyzing the JCL for syntax errors and to generate reports, JOB/SCAN may need to read the user catalog, the load libraries, control card libraries, and write to output files. The Security Exit is called prior to opening the file to verify that the user running JOB/SCAN has the proper access authority.

If access privilege is denied to a file, the exit returns a non-zero return code (in register 15) and no attempt is made to access the dataset. A message indicating that access has been denied is displayed to the ISPF panels or in the Structured JCL Listing (SJL). The JOB/SCAN run will continue. Therefore, if a message is received indicating that access is denied, note the file name and verify that the expected results were received.

### DSNNAME Validation

DSNNAME Validation validates the access to all datasets within the job stream being scanned by JOB/SCAN. To use this function, code the DSNNAME parameter on the Security Exit Run Option.

With DSNNAME Validation enabled, JOB/SCAN calls the Security Interface and passes it the USER and/or GROUP parameters that were found on the JOB statement or //\*LOGONID statement in the job currently being scanned.

This call is made immediately after the JOB card or //\*LOGONID is scanned by JOB/SCAN. In this call, a blank dataset name is passed. If the USER and/or GROUP specified on the scanned job is different than the USER and/or GROUP of the JOB/SCAN job or TSO session, then the Security Exit must be running APF-authorized in order to perform the DSNNAME validation function. Check with the Security Administrator or Systems Programmer.

If you are running APF authorized in foreground under TSO (in ISPF or ISPF Edit macro), link the security exit as RENT. The security exit name must be placed in table AUTHTSF in member IKJTSO00 of SYS1.PARMLIB. If there is no USER or GROUP specified on the JOB statement of the job being scanned, JOB/SCAN omits this call. If the exit returns a non-zero return code (in register 15) from this call, this message is displayed in the Structured JCL Listing (SJL):

```
DSS4759A DSNNAME SECURITY VALIDATION SUSPENDED FOR THIS JOB
```

DSNNAME validation occurs after JOB/SCAN processes a DD statement from the JCL input stream that contains a dataset name. The expected access of Read, Update, or Alter is determined from the DISP parameter specified in the DD statement. This can be overridden by the I/O code (+I, +O) in the JCL comment field or from the Short Description file. If the exit returns a non-zero return code (in register 15) from this call, this message is displayed in the Structured JCL Listing (SJL):

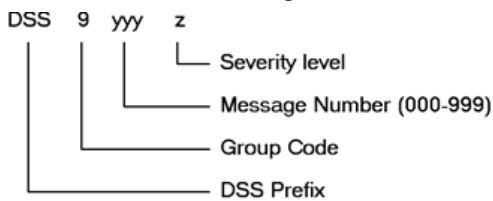
```
(READ )
DSS4757E (UPDATE) ACCESS TO DATA SET DENIED TO USER userid
```

(ALTER )

The DSNNAME validation function allows the user exit to pass a message back to JOB/SCAN to be displayed in the Structured JCL Listing. This message consists of a three-digit numeric value, which becomes part of the standard message number format and the text that makes up the short message. The length of this message cannot exceed fifty-two characters. A severity code associated with the message is also passed back to JOB/SCAN. This severity code affects the return code from the current execution of JOB/SCAN, as well as whether JOB/SCAN aborts processing.

Messages passed back to JOB/SCAN follow the same format as other JOB/SCAN messages. A group code of 9 identifies these messages as being from an exit with the format shown in .

#### Format of DSNNAME Messages



Message Component	Description
Group Code	User exit messages are always identified with a Group Code of 9. This value is assigned by JOB/SCAN and cannot be changed by the exit.
Message Number	Three-digit numeric number that identifies the message as unique. The value is coded in the first three bytes of the message text by the exit routine. If no value is coded, JOB/SCAN uses 999 when displaying the message.
Severity Indicator	This is based upon the severity code that the Security Exit passes back to JOB/SCAN. The severity indicator is translated as follows:

Severity Indicator	Severity Definition	Severity Code
A	Advisory message	0
W	Warning message	1-4
E	Error message	5-19
C	Catastrophic (JOB/SCAN stops)	20-4096

★ If the Suppress Error Messages field on the Processing Options panel contains a value other than NO, the error suppression indicated by this field applies to error messages generated by the Security Exit as well as those generated by JOB/SCAN.

### JOB/SCAN Termination

Termination notifies the Security Exit that JOB/SCAN is terminating. This allows the Security Exit to free allocated storage and perform any termination routines.

### Modifying a Security Exit Program


If the site has special requirements for a security exit, the shipped exit programs may be modified to accommodate them. If you are running APF-authorized in foreground under TSO (in ISPF or ISPF Edit macro), link the security exit as REENTRANT.

### Security Exit Modules

ASG provides several security exit programs to use with JOB/SCAN. In addition to the load modules in the load library, the source code for these programs is included in the SJJSSAMP library. Installations with special security exit requirements can modify the programs as needed.

Security Exit Module	Description
ACF2EXIT	Security exit program for interfacing with the CA-ACF2/MVS security package (provided for compatibility with security interfaces established for JOB/SCAN releases prior to 5.4). Does not support the dataset name validation function DSNAME (see the SAFx SJJSSAMP members for DSNAME support).
RACFEXIT	Security exit program for interfacing with the RACF and CA-TOP SECRET security packages (provided for compatibility with security interfaces established for JOB/SCAN releases prior to 5.4). Does not support the dataset name validation function DSNAME (see the SAFx SJJSSAMP members for DSNAME support).
SAFO	Security exit program that uses the System Authorization Facility (SAF) to interface with RACF, CA-ACF2/MVS, and CA-TOP SECRET. <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>★ This program should not be used when running APF-authorized under ISPF with <i>DSNAME</i> coded in the Security Exit Run Option. It issues a RACINIT macro, and there is no way to specify a subpool for the ACEE.</p> </div>
SAF1	Security exit program that uses the third-party RACHECK instead of the RACINIT/RACHECK used by SAFO. <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>★ This program works only with RACF Release 1.8 or greater, or the equi-</p> </div>



Security Exit Module	Description
SAF2	<div data-bbox="488 268 1484 506" style="border: 1px solid purple; padding: 10px; margin-bottom: 10px;"> <p> valent CA-ACF2/MVS or CA-TOP SECRET releases. SAF1 uses subpool 251 for storage when invoked in an APF-authorized environment under ISPF. (This is because subpool 0 storage is deleted after the module is executed, while subpool 251 storage is not deleted until the job step task terminates.)</p> </div> <p>Security exit program that performs the same as SAF1, except that it uses the STATUS=ACCESS parameter of the RACROUTE macro to eliminate logging of RACF violations (when the user doesn't have proper access authority). CA-ACF2/MVS (ACF2 6.1 gl 9510) and CA-TOP SECRET (TSS 4.4 gl 9506) also support the STATUS=ACCESS parameter.</p> <p>CA-ACF2 requires APF authorization to use <i>STATUS=ACCESS</i>, unless a (CA-ACF2) SAFDEF record is defined with the NOAPFCHK keyword. ASG recommends coding NOAPFCHK.</p>

## Security Interface Description

### Registers Used by the Exit Program

Each time the security interface is called, the registers are set as described in this table:

Register	Contents
1	+0 Address of the Security Interface Parameter (SIP) list. +4 Address of Exit Member name. +8 Address of User Exit communication name.
13	Address of a standard MVS register save area.
14	Return address.
15	Entry point address of the security interface exit routine.

## Program Requirements

### To set up the security interface

1. After entering the exit routine, save all register contents.
2. Before returning control to the calling routine, restore all registers but Register 15, and ensure that Register 15 contains one of these return codes:
  - Zero: JOB/SCAN issues no messages.
  - Non-zero: JOB/SCAN issues message DSS4755, if the reason for the call was for an access control function.
3. JOB/SCAN issues message DSS4759, if the reason for the call was DSNAME and the dataset name field passed was blank.
4. JOB/SCAN issues message DSS4757, if the reason for the call was DSNAME validation and the dataset name field passed was not blank.

### Security Interface Parameter List

Offsets Dec (Hex)	Type	Length	Name	Description
0 (00)	Character	4	SIPID	Control block ID ('SIP').
4 (04)	Address	4	SIPSADDR	Virtual address of SIP.
8 (08)	Character	1		Reserved field.
				Reason security exit is being called. These values are used for JOB/SCAN access control:
				C'CATALOG REA' - Read the user catalog.
				C'DYN ALLOC ' - Allocate a library needed by JOB/SCAN.
				C'EDIT REQST ' - Edit the dataset using ISPF Edit.
				C'IMS LIBRARY' - Read dataset for IMS processing.
9 (09)	Character	11	SIPRSN	C'JCLLIB READ' - Read dataset to obtain cataloged procedures and INCLUDE libraries.
				C'MBR SELECT ' - Read JCL dataset.
				C'READ CC MBR' - Read dataset for control card processing and PDS member verification.
				C'STEPLIB REA' - Read dataset for PGM verification.
				C'VTOC READ' - Read VTOC for dataset validation.
				C'DSN AUTH CK' - JOB/SCAN DSNAME Validation:

Offsets				
Dec (Hex)	Type	Length	Name	Description
				<p>Validate the expected access to the dataset currently being scanned by JOB/SCAN. If the dataset field SIPDSN is blank, validate the USER and/or GROUP name fields, SIPUSRID and SIPGRPID.</p> <p>C'TERMINATION' - JOB/SCAN has completed validation and is terminating.</p>
20 (14)	Character	44	SIPDSN	Dataset name to be used for validation. This field is blank for the termination call, and is also blank when it is called by JOB/SCAN to validate the USER and/or GROUP parameters from a JOB card JOB/SCAN is scanning.
64 (40)	Character	6	SIPVOL	Volume serial number of the dataset, if available.
70 (46)	Character	1	SIPTYPE SIPREAD SIPUPDAT SIPALTER	<p>Type of access desired:</p> <p>C'R' -Read access.            C'W' -Read and write access.            C'A' -Alter access.</p>
71 (47)	Character	1	SIPDVTYP SIPDASD SIPTAPE SIPUNKWN	<p>Device type:</p> <p>C'D' -DASD device.            'T' -Tape device.            'U' -Unknown device.</p>
72 (48)	Character	8	SIPUSRID	For DSNAMES validation only. Contains USER field from job card of job JOB/SCAN is scanning.
80 (50)	Character	8	SIPGRPID	For DSNAMES validation only; contains GROUP field from job card of job JOB/SCAN is scanning.
88 (58)	Fixed	4	SIPUSERX	Reserved for exit's use. This field is initialized to binary zeros when exit is initially loaded and not modified further; it is to be used exclusively by the exit routine.
92 (5C)	Fixed	2	SIPSEVCD	<p>For DSNAMES validation only. This field contains severity code (0, 4, 8, 20) of message (SIPMSG) to be written by JOB/SCAN:</p> <p>H'0' -Advisory message            H'4' -Warning message            H'8' -Error message            H'20' -Catastrophic message</p>

Offsets Dec (Hex)	Type	Length	Name	Description
94 (5E)	Character	52	SIPMSG	For DSNAMES validation only. This field holds a 3-byte message number followed by a message to be placed in Structured JCL Listing.
146 (92)	Fixed	2		Reserved field.
148 (94)	Character	8	SIPPGMNM	For DSNAMES validation only. This field contains the program name for EXEC PGM= statement of the current step JOB/SCAN is scanning.
156 (9C)	Character	8	SIPJOBNM	For DSNAMES validation only. This field contains job name for Job statement of current job JOB/SCAN is scanning.
164 (A4)	Character	20	SIPPRGMR	For DSNAMES validation only. This field contains programmer name field that was coded on Job statement of the current job JOB/SCAN is scanning.
184 (B8)	Character	144	SIPACCTG	For DSNAMES validation only. This field contains the accounting field that was coded on the Job statement of the current job JOB/SCAN is scanning.
328 (148)	Character	8	SIPDD	For DSNAMES validation only. This field contains name of DD statement that is associated with the dataset name.
336 (150)	Character	8	SIPSECLB	For DSNAMES validation only. This field contains SECLABEL parameter that was coded on JOB statement of the current job JOB/SCAN is scanning.
344 (158)	Address	4	SIPUSERN	Address of the User Exit communication word.

## Other Interfaces

This topic provides detailed information about the installation of the other interfaces that are compatible with JOB/SCAN.

- [CA-PANVALET Interface](#)
- [Running Jobs in an APF-Authorized Environment](#)
- [CA-1 \(UCC1\) Tape Management System](#)
- [ChangeMan Interface](#)
- [DB2 Interface](#)
- [ENDEVOR Interface](#)
- [IMS \(Information Management System\) Interface](#)
- [CA-DYNAM/TLMS Interface](#)
- [ZARA Interface](#)
- [DFSMSrmm – Removable Media Manager](#)
- [CA-11 Interface](#)
- [CONTROL-R Interface](#)
- [SCLM Interface](#)

### CA-PANVALET Interface

To enable JOB/SCAN to interface with CA-PANVALET libraries, use the installation procedure that corresponds with the CA-PANVALET release level used at your site.

#### For CA-PANVALET Release 12 or later

If the library that contains PAM (a CA-PANVALET load module) is not in the link pack area or in the link list, add it to the STEPLIB or SJJ SPLIB. Then,

- If executing JOB/SCAN in batch mode, modify the STEPLIB in JS6PROC.

Or

- If executing JOB/SCAN in ISPF mode, modify the TSO logon PROC, by adding the library containing the PAM load module to the load library concatenation list.

### Running Jobs in an APF-Authorized Environment

Certain optional functions of JOB/SCAN require an APF-authorized environment. These are the functions:

- Verifying the SCHENV JOB statement parameter. JOB/SCAN invokes the Workload Management Scheduling Environments Determine Execution Service to determine if the scheduling environment exists and is available to the current system.
- Verifying JES2/JES3 Remote Destinations when running z/OS prior to V1R9.

- Verifying access to datasets using the USERID parameter from the JCL being scanned when it is different than the USERID of the JOB/SCAN job or TSO session. Refer to the [Security Interface Overview](#) for more information.
- To query a volume within the tape storage environment. For example, a DD statement refers to an uncataloged dataset and the VOL=SER= parameter is specified but the Unit= parameter is omitted, JOB/SCAN will query the Library Control System (LCS) to see if this volume is in a tape library. If running SMS:
  - If the DFSMS subsystem uses the USERID in the ACS routines to determine SMS constructs, and the USERID in the JCL being scanned is different than the USERID of the JOB/SCAN job or TSO session.
  - For non-APF, JOB/SCAN ignores the USERID of the JCL being scanned, and the USERID of the JOB/SCAN job or TSO session is used, possibly causing unpredictable results.

### Setting Up the ISPF Environment

JOB/SCAN can perform the APF-authorized functions under ISPF if TSO/E Release 2 is installed on the user system.

#### **To run APF-authorized in an ISPF environment**

1. Rename these load modules:

- J00YAAI
- J00YAAIU
- J00YAAIA
- J00YAAI

2. Move module J00YAAI to an APF-authorized dataset.

If the dataset is not in the LNKLIST or in LPA, it must be added to the STEPLIB concatenation in the TSO Logon PROC. It will not be loaded from the SJJSLMOD allocation. The Systems Programmer might need to perform this function.

3. Add the name J00YAAI to the SYS1.PARMLIB member IKJTSOxx under sections AUTHPGMS and AUTHTSF. The Systems Programmer may need to perform this function.



The default setting of the ISPF SPECIFICATIONS Run Option is AUTHFUNCTIONS, allowing JOB/SCAN to invoke APF-authorized functions. (To by-pass APF-authorized functions and improve system performance, set the Run Option to NOAUTHFUNCTIONS.)

### APF Authorization For Batch Processing

JOB/SCAN runs in an APF-authorized environment by following the MVS authorization rules. These are the rules:

- All load modules must reside in APF-authorized libraries. The JOBLIB/STEPLIB DD concatenation may specify only APF-authorized libraries in the JOB/SCAN JCL.
- The program being executed must be link-edited with an Authorization Code of 1. This may be accomplished by renaming these load modules

- J02YJS
- J02YJSA
- J02YJSU
- J02YJS

JOB/SCAN checks the ddname JLLOD for exits such as the Selection Exit, Report Exit, Standards Exit, and Security Exit. If these exits are found in an APF-authorized library, they will run APF-authorized; otherwise, they run unauthorized.

### APF Authorization For Security Exit Users

If using the JOB/SCAN Run Option called SECURITY EXIT=xxxx,DSNAME, and checking JCL that executes under a USERID that is not the USERID of the JOB/SCAN job or current TSO session, be sure the security exit named xxxx is in an APF-authorized library. If you are running APF-authorized in foreground under TSO (in ISPF or ISPF Edit macro), link the security exit as RENT. The security exit name must be placed in table AUTHTSF in member IKJTSO00 of SYS1.PARMLIB.

The security exit provided for this function is named SAF1. This exit works with:

- RACF Release 1.8 or later
- Equivalent CA-ACF2/MVS Release 6
- Equivalent CA-TOP SECRET

Security exits are invoked APF-authorized under these conditions:

- **Batch Processing** Perform the steps described in [APF Authorization For Batch Processing](#). Make sure the Security Exit resides in an APF-authorized library.
- **ISPF Processing** Perform the steps described in [Setting Up the ISPF Environment](#). Make sure the Security Exit resides in an APF-authorized library, is link edited with the RENT option, and the security exit name is in the AUTHTSF member of SYS1.PARMLIB.

### The Security Exit Module

Security exit module SAF1 is supplied by ASG. It is shipped as source code in the SJJSSAMP library, as well as a load module in the SJJSLMOD library. SAF1 is written to use the third-party RACHECK instead of the RACINIT/RACHECK done by SAF0. It uses subpool 251 for its storage if it is invoked APF-authorized because, under ISPF, subpool 0 storage is deleted after the module is executed. Subpool 251 storage is not deleted until the job step task terminates. This is a restriction of ISPF's APF-authorization processing. This exit works only with RACF Release 1.8 or greater, the equivalent of CA-ACF2/MVS, or the equivalent of CA-TOP SECRET releases.

## CA-1 (UCC1) Tape Management System

JOB/SCAN can access the CA-1 (UCC1) Tape Management System (TMS) catalog to obtain information about tape datasets to verify the volume serial number coded in the JCL or from the system catalog against the TMS catalog. After the CA-1 interface is installed, JOB/SCAN automatically invokes it as necessary.

### Establishing the Interface for CA-1

#### *To establish the interface*

1. Identify the TMS Library.

Make sure the TMS interface routine (TMSSVC) is available during JOB/SCAN execution using either of these methods:

- Make sure TMSSVC is link-listed.
- Add the TMS load library name to the STEPLIB concatenation in the logon PROC (for foreground processing) and in the JOB/SCAN batch PROC (JS6PROC).

## 2. Turn On the Interface.

Turn on the interface by adding one of these parameters to the Program Product Interface Run Option in the appropriate SJJSPARM member:

- If the CA-1 release is 5.0 or greater, use CA1R5.
- Or**
- If the CA-1 release is prior to Release 5, use CA1.

For example:

```
PROGRAM PRODUCT INTERFACE =CA1R5
```

The CA-1 Tape Management System is invoked automatically each time JOB/SCAN is run on a job containing tape datasets.

## ChangeMan Interface

ASG provides a ChangeMan interface which makes successful JOB/SCAN Validation a prerequisite to promoting JCL and cataloged procedures. The interface supports complete JCL Validation, including compliance with in-house standards.

### Establishing the Interface for ChangeMan

#### *To establish the interface*

##### 1. Set up ChangeMan Types for JCL and PRC.

ChangeMan supplies Types JCL and PRC. The processing for these types is fixed as copy only. Set up new Types such as JCS and PRO, for JCL and PROCs. The LIKE field in the Library Type Administration panels must be blank. Invoke the CMN\$\$JCS and CMN\$\$PRO skeletons for the new Types.

##### 2. Copy the Skeletons to SJJSSLIB.

- a. Copy the three ChangeMan skeletons CMN\$\$JCS, CMN\$\$PRO and CMN\$\$SCN from SJJSSAMP into an ISPF skeleton library available to ChangeMan (for example, a PDS allocated to DDNAME SJJSSLIB during a ChangeMan session).
- b. If the skeletons supplied by ChangeMan are modified, modify the existing skeletons rather than using the CMN\$\$JCS and CMN\$\$PRO versions provided by ASG. Part 2 of each skeleton contains the required ASG modifications.



- c. CMN\$\$SCN is a ASG-supplied skeleton containing JOB/SCAN JCL which is )IMbedded in CMN\$\$JCS and CMN\$\$PRO. It may be renamed if the )IM CMN\$\$SCN lines are changed in the )IMbedding members.

### 3. Modify CMN\$\$SCN.

The library specified in DD STEPLIB is the name of the JOB/SCAN load library. This statement may be deleted if JOB/SCAN is installed in a link listed library.

The library specified in DD JLRCL is the library containing member JSOPT01, JSOPT02, and JSOPT03 using your site-specific JOB/SCAN SJJSPARM members. See this JLRCL example:

```
//JLRCL DD DISP=SHR,DSN=?YOUR.SITE'S.JOBSCAN.SJJSPARM?(?JSOPT01)
// DD DISP=SHR,DSN=?YOUR.SITE'S.JOBSCAN.SJJSPARM?(?JSOPT02)
// DD DISP=SHR,DSN=?YOUR.SITE'S.JOBSCAN.SJJSPARM?(?JSOPT03)
```

The library specified in DD JLMSG is the library containing DSSMSGnn members, and is usually the same as JLRCL. Do not specify a member name for JLMSG.

The STANDARDS NAME statement is used when a JOB/SCAN standards exit needs to be executed. Specify the exit name to make standards validation a prerequisite to successful JCL promotion. If site-written standards do not need to be enforced, remove the statement.

### LIBDEF CLIST Example

The figure shows an example showing the ISPF LIBDEF COMMANDS inserted into the JEM CLIST:

#### ChangeMan interface LIBDEF commands

```
PROC 0 DJSNAME (DSSI.$JSnnn)
.
.
.
CONTROL END(ENDO) NOMSG NOFLUSH
/*CONTROL END(ENDO) MSG LIST CONLIST SYMLIST NOFLUSH /*S80725*/
ISREDIT MACRO (PARMSTR)
ALLOC FI (JLLOD) DA ('&DJSNAME..SJJSLMOD') SHR REU
ISPEXEC LIBDEF SJJSLMOD DATASET +
ID ('&DJSNAME..SJJSLMOD') STACK
ISPEXEC LIBDEF SJJSPLIB DATASET +
ID ('&DJSNAME..SJJSPLIB') STACK
ISPEXEC LIBDEF SJJMLIB DATASET +
ID ('&DJSNAME..SJJMLIB') STACK
.
.
.
ENDING: +
IF &RCLALC = 'N' THEN FREE DDN(JLRCL)
ISPEXEC LIBDEF SJJSLMOD
ISPEXEC LIBDEF SJJSPLIB
ISPEXEC LIBDEF SJJMLIB
FREE F(JLLOD)
EXIT
```

If the JOB/SCAN libraries are allocated using the ISPF LIBDEF service, the STACK parameter must be added for correct operation of the Edit macros under ChangeMan edit.

## DB2 Interface

ASG provides a DB2 interface for extracting additional information for JOB/SCAN to validate. The information is available for batch DB2 jobs that access databases through the DSN RUN command.

### JOB/SCAN Validation

During its run, JOB/SCAN:

- Parses and validates syntax of the DSN command and parameters, flagging any errors.
- Verifies that the DB2 application program identified in the DSN RUN command exists.
- Verifies that the DB2 library name specified in the DSN command exists.
- Verifies that the DB2 subsystem specified in the DSN command is an active MVS subsystem.

JOB/SCAN does not validate DB2 utility commands.

### Establishing the Interface for DB2

#### *To establish the interface*

1. Turn On the DB2 Interface.

Turn on the interface by adding the DB2 parameter to the Program Product Interface (PPI) Run Option in the appropriate SJJSPARM member.

For example:

```
PROGRAM PRODUCT INTERFACE =DB2
```

2. Code the Control Table Entries.

Make sure the Control Table entries are coded in the appropriate SJJSPARM member. (See [Control Tables](#) for details of Control Table AB.)

For example:

```
.....+.....1.....+.....2.....+.....3.....+.....4
ABSYSTSIN      IKJEFT01
ABSYSTSIN      IKJEFT1A
ABSYSTSIN      IKJEFT1B
```

The DB2 interface will be invoked automatically each time JOB/SCAN encounters a job that accesses a DB2 database through the DSN RUN command.

### Sample Batch DB2 Job Stream

This is a sample batch DB2 job stream:

```
//STEP EXEC PGM=IKJEFT01
//      :
//      :
//SYSIN DD *
DSN  SYSTEM(DB2P)
RUN  PROGRAM(PAYDB)  LIB (' PROD.SJJSIMOD' )  PLAN (PAYDB)
END
```



JOB/SCAN does not validate DB2 Utilities.

## ENDEVOR Interface

ASG provides an ENDEVOR interface which makes JOB/SCAN function as a part of promoting JCL and cataloged procedures. The interface supports complete JOB/SCAN JCL validation, reformatting, changing, and standards checking.

### Establishing the Interface for ENDEVOR

#### *To establish the interface*

1. Copy the Model Processor Source Code into ENDEVOR .
2. Copy the ENDEVOR model processor source code member from SJJSSAMP into the dataset containing the ENDEVOR processor source.
3. Modify the Processor Source Code.

Steps have been included in the model processor source code for many ASG software functions. Delete those steps which execute functions unwanted in the promotion process.

The library specified in DD STEPLIB is the name of the JOB/SCAN load library. This statement may be deleted if JOB/SCAN is installed in a link listed library.

The library specified in DD JLRCL is the library containing JSOPTnn JOB/SCAN initialization members.

The library specified in DD JLMSG is the library containing DSSMSGnn members, and is usually the same as JLRCL. Do not specify a member name.

4. Build an ENDEVOR processor using the revised model processor source code to execute as part of the promotion process. This step depends upon how the site has installed ENDEVOR.

## IMS (Information Management System) Interface

ASG provides an Information Management System (IMS) interface for extracting additional information for JOB/SCAN to validate. The information is available for jobs that access databases through the DLI, BMP, and DBB functions. IMS access of DB2 databases is also supported.

### JOB/SCAN Validation

During its run, JOB/SCAN validates that:

- The IMS application program identified in the PARM parameter of the EXEC statement exists.
- The PSB specified in the Parm field and its associated DBDs can be found in the libraries specified by ddname //IMS for DLI-type processing or ddname //IMSACB for DBB-type processing.
- For BMP processing, the ACB specified in the Parm field is found in the ACB library.

## Establishing the Interface for IMS

### To establish the interface

1. Include the File Allocation Statement.

Define the IMS ACB library in the File Allocation section of the appropriate SJJSPARM member (BMP processing only). For example:

```
ACBLIB =DSN=PROD.IMS.ACBLIB,SHR
```



JOB/SCAN performs special validation for batch jobs that access IMS databases using the DFSRRC00 driver with the DLI, BMP, and DBB options and the DFHDRP driver that accepts keyword parameters in the Parm field.

2. Turn On the IMS Interface.

Turn on the interface by adding the IMS parameter to the Program Product Interface Run Option in the appropriate SJJSPARM member. This is an example for IMS:

```
PROGRAM PRODUCT INTERFACE =IMS
```

This is an example for IMS with access to DB2 databases:

```
PROGRAM PRODUCT INTERFACE =IMS,DB2
```

This is an example for IMS with the optional IMS subsystem ID specified:

```
PROGRAM PRODUCT INTERFACE =(IMS,X16H)
```

where X16H is a 1-to-4-character subsystem ID specified at IMS SYSGEN time.

3. Code the Control Table Entries.

Make sure the appropriate Control Table entries are coded in the SJJSPARM members. See [Control Tables](#) for details on Control Tables AD and AP for IMS, and Control Table AB for IMS access of DB2 databases.

This is an example for IMS:

```
.....1.....2.....3.....4
  ADDFSRRC00      UR
  APDFSRRĊ00      2
```

This entry is also required for IMS/DB2:

```
.....1.....2.....3.....4
  ABDDITV02      DFSRRC00
```

The IMS interface is invoked automatically whenever JOB/SCAN encounters the IMS driver program accessing databases through the DLI, BMP, or DBB functions.

## CA-DYNAM/TLMS Interface

JOB/SCAN can access the CA-DYNAM/TLMS tape management system Volume Master File (VMF) to verify that volume and dataset information coded in the JCL or obtained from the system catalog matches what is in the VMF. After the TLMS interface is installed, JOB/SCAN automatically invokes it as necessary.

Choose from one of these two interfaces for TLMS:

- As a PPI option that directly reads the VMF
- As a standards program.

### Establishing the Interface for the PPI Option

#### To establish the interface

1. Include the File Allocation statement.

Define the TLMS VMF in the FILE ALLOCATION section of the appropriate SJJSPARM member. For example:

```
JTLMSVMF=DSN=TLMS.VMF, SHR
```

2. Turn on the TLMS Interface by adding the TLMS parameter to the Program Product Interface Run Option in the appropriate SJJSPARM member.

For example:

```
Program Product Interface=IDCAMS, HSM, TLMS
```

### Establishing the Interface for the Standards Program

#### To establish the interface

1. Identify the TLMS Volume Master File (VMF), VMF index, Retention Master File (RMF), and Alternate Log Backup.
2. Add DD statements to JOB/SCAN as shown in the figure.

#### DD statements added for the CA-DYNAM/TLMS interface

```
//STEPLIB DD DSN=CAI.CAILIB, DISP=SHR
//TLMSOPTS DD DSN=CAI.PPOPTION(TLMSIPO), DISP=SHR
//SORTLIB DD DSN=SYS1.SORTLIB, DISP=SHR
//CAIRMF DD DSN=CAI.TLMS.RMF, DISP=SHR
//CAIVMF DD DSN=CAI.TLMS.VMF, DISP=SHR
//CAIVMFI DD DSN=CAI.TLMS.VMF, DISP=SHR
//CAIVMFS DD DSN=CAI.TLMS.VMF, DISP=SHR, DCB=(BUFNO=10, OPTCD=C)
//CAIVMFXI DD DSN=CAI.TLMS.VMFINDEX, DISP=SHR
//BKUP DD DSN=CAI.TLMS.BKUPALOG, DISP=SHR
//SYSPRINT DD UNIT=SYSDA, SPACE=(TRK,(1,1)), DCB=(BLKSIZE=1330, LRECL=133, RECFM=FB)
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//ALLSTOR DD UNIT=SYSDA, SPACE=(CYL,(5,5)), DCB=BLKSIZE=2150
//USEDSTOR DD UNIT=SYSDA, SPACE=(CYL,(5,5)), DCB=BLKSIZE=2150
//RPTSFIL DD UNIT=SYSDA, SPACE=(CYL,(10,2)), DCB=BLKSIZE=2150
//SORTFILE DD UNIT=SYSDA, SPACE=(CYL,(10,2)), DCB=BLKSIZE=2150
//SORTWK01 DD UNIT=SYSDA, SPACE=(CYL,(10))
//SORTWK02 DD UNIT=SYSDA, SPACE=(CYL,(10))
//SORTWK03 DD UNIT=SYSDA, SPACE=(CYL,(10))
//SYSIN DD UNIT=SYSDA, SPACE=(TRK,(1,1)), DCB=(BLKSIZE=4080, LRECL=80, RECFM=FB)
```

3. Activate the Interface by coding the Exit Name Run Option statement in the appropriate SJJSPARM member as follows:

```
STANDARDS NAME=J00XTLMS
```

## ZARA Interface

JOB/SCAN can access the ZARA database to obtain information about tape datasets to verify the volume number coded in the JCL or from the system catalog against the ZARA database. After the ZARA interface is installed, JOB/SCAN automatically invokes it as necessary.

### *To establish the interface*

1. Identify the LOADLIB containing the API program ZARAAPI1.
2. Make sure the ZARAAPI1 interface routine is available during JOB/SCAN execution using either of these methods:
  - Make sure ZARAAPI1 LOADLIB is link-listed. Add the ZARAAPI1 load library name to the STEPLIB concatenation in the logon PROC (for foreground processing) and in the JOB/SCAN batch PROC (JS6PROC). Turn on the ZARA interface. See [Program Product Interface \(PPI\)](#).

**Or**

- Add the ZARA parameter to the Program Product Interface Run Option in the appropriate SJJSPARM member. For example:

```
PROGRAM PRODUCT INTERFACE =ZARA
```

JOB/SCAN invokes the interface automatically each time JOB/SCAN is run on a job containing tape datasets.

## DFSMSrmm – Removable Media Manager

JOB/SCAN accesses IBM's RMM Control Data Set (CDS) to obtain information regarding tape datasets known to RMM. This information can be used with information from the system catalog and information on the tape DD card to verify that the DD card, the Catalog, and RMM's knowledge of the physical tape are consistent.

Information checked is VOLSER, tape dataset name, availability (out in a disaster store location or in the library), dataset sequence, status (scratch or master file), etc. JOB/SCAN automatically invokes the RMM Applications Interface (API) when required for a tape dataset.

### Requirements

These conditions must be met to utilize the JOB/SCAN DFSMSrmm interface:

- DFSMS level V1R3 or higher installed.
- For V1R3, you must be at maintenance level 9806 or have PTF UW47006 applied. If running V1R4, you must be at maintenance level 9806 or have PTF UW47007 applied. For V1R5 and above, no special maintenance is needed.



The best way to verify the maintenance level is to locate EDGDOCX in SYS1.SJSSAMP and EDGXAPI in SYS1.LINKLIB. If these members are present, you have the correct maintenance level.

- Make sure that RMM is operative.
- Make the DFSMSrmm load library available to both JOB/SCAN and the JS6PROC.

## Establishing the Interface for DFSMSrmm

### To establish the interface

1. Identify the LOADLIB containing the API program EDGXAPI, and make it accessible to JOB/SCAN.
  - a. Make sure the EDGXAPI interface routine is available during JOB/SCAN execution using either of these methods:
  - b. Make sure EDGXAPI LOADLIB is link-listed, if it is not in SYS1.LINKLIB. Add the EDGXAPI load library name to the STEPLIB concatenation in the logon PROC (for foreground processing) and in the JOB/SCAN batch PROC (JS6PROC), if EDGXAPI is not in SYS1.LINKLIB. Turn on the RMM interface.
2. Turn on the interface by adding the RMM parameter to the Program Product Interface Run Option in the appropriate SJJSPARM member. For example:

```
PROGRAM PRODUCT INTERFACE =RMM
```

RMM (Removable Media Manager) interface will be invoked automatically each time JOB/SCAN is run on a job containing tape datasets. If the dataset names are not catalogued, the unit name in the DD card must indicate that this is a tape dataset.



If you have any Stor or Loan locations defined to RMM which are to be considered as on-site, add appropriate entries to the Control Table AM entries.

## CA-11 Interface

JOB/SCAN automatically detects the presence of a CA-11 rerun manager step in the JCL, and simulates its functions. Ensure that a Control Table CE Entry is present containing the program name U11RMS.

## CONTROL-R Interface

JOB/SCAN automatically detects the presence of a CONTROL-R rerun manager step in the JCL, and simulates its functions. No action is necessary at either JOB/SCAN installation time or JOB/SCAN run time.

## SCLM Interface

The SCLM Interface is a sample REXX EXEC that runs JOB/SCAN interactively. The SCLM Promote Verification process calls this EXEC.

### Establishing the Interface for SCLM

#### *To establish the interface*

1. Modify EXECJS1 REXX EXEC to reflect your site's SCLM project definitions and JOB/SCAN allocation requirements. The EXECJS1 REXX EXEC is found in the SJJSCST library.
2. Modify your SCLM project definition to invoke JOB/SCAN.
3. Build test JCL.
4. Test your JCL through the SCLM promote process using the new project definition.
5. After successful testing of the JOB/SCAN verification process, replace the production project definition member with the new member, which invokes JOB/SCAN.

### Modify Your SCLM Call

#### *To modify your call from SCLM*

1. Work with the SCLM Administrator to create a new SCLM project definition member.

This will be modified to invoke the JOB/SCAN verification process. (Locate a current SCLM project definition member and create a new copy using the existing SCLM project definition member as a model).

2. Find the current name of the project definition member, referred to as the PRJ\_DEF parameter, and change it to the new project definition name.
3. Add these fields within the FLMCNTRL macro statement in the \*\*.PROJDEFS.SOURCE member containing the Project definitions:

```
PRMEXT1=EXECJS1,PEXT1CM=TSOLNK,PEXT1DS=MY.DSSI.EXEC.LIB
```

4. Assemble and link the new project definition in SCLM. Ensure that the JOB/SCAN REXX EXEC, EXECJS1, is moved into the library specified in PEXT1DS step 3.
5. Modify the EXECJS1 REXX EXEC so that the appropriate parameters are passed to JOB/SCAN. Possible changes include:
  - TYPE - Refers to the suffix of the input library. This should be changed to match the value of your JCL library.
  - D99NDSN1 - This sets up the entire input library name. If you do not use the IBM default of project.group.type, you should change this name.
  - EXECJS1 - Returns a return code back to SCLM. Because SCLM will not promote anything with a return code greater than 0, EXECJS1 sets the return code to 0 if the return code from JOB/SCAN is 4 or less.



For more specific details on setting up the SCLM Promote User Exit Routine, refer to IBM's *ISPF SCLM Project Manager's Guide*, SC28-1319.



For EXECJS1 to call JOB/SCAN, it must be invoked from a valid ISPF environment and all of the JOB/SCAN libraries must be allocated using either an allocation EXEC or in the LOGON PROC.\*

## Extended Parameter List Samples

This topic provides samples of the contents of each data element in the Extended Parameter List. These data elements are used when writing site-specific standards. The complete Extended Parameter List can be found in the SJSSTDL (CEPL).

Each data element section is described in this table. These sections correspond to the sections of the Extended Parameter List and reflect the order in which the data elements are to be defined.

Data Element	EXIT PARM	Field-naming Prefix	Description
JOB Data	EPL-JOB-DATA	EPLJ-	These fields pass JOB statement information from JOB/SCAN to the standards program.
SET Data	EPL-SET-DATA	EPLS-	These fields pass SET statement information from JOB/SCAN to the standards program.
JCLLIB Data	EPL-JCLLIB-DATA	EPLB-	These fields pass JCLLIB statement information from JOB/SCAN to the standards program.
JES2 Data	EPL-JES2-DATA	EPL2-	These fields pass JES2 statement information from JOB/SCAN to the standards program.
Error Message Data	EPL-ERROR-DATA	EPL-	This table holds up to 50 messages per JCL statement, with each table entry specifying the severity level and error text level indicator associated with the error message. This information is returned to JOB/SCAN after all JCL standards verifications are performed for the current statement. These messages are included in the Structured JCL Listing.
Global Control Data	EPL-CONTROL-DATA	EPLC-	This data element comprises eight types of fields: <ul style="list-style-type: none"> <li>Value set by JOB/SCAN</li> </ul>

Data Element	EXIT PARM	Field-naming Prefix	Description
			<p>specifying entity level currently being validated.</p> <ul style="list-style-type: none"> <li>• Severity level of error detected by the standards program.</li> <li>• Standards program error message generated when an error occurs. Fields 3, 4, and 5 are used to update the error table that passes the error messages to JOB/SCAN.</li> <li>• Indicator that determines whether the extended text for this error message appears in the Structured JCL Listing.</li> <li>• Indicator of JOB/SCAN invocation method: ISPF; batch; Edit macro.</li> <li>• Group of fields used as switches set by the standards program, indicate the call sections to be invoked within the program.</li> <li>• Index number of the current JCL entity. This index number is assigned by JOB/SCAN when validating the JCL and is made available to the standards program.</li> <li>• Field containing up to 32 characters of parameter information being passed to the standards program</li> </ul>

Data Element	EXIT PARM	Field-naming Prefix	Description
			from JOB/SCAN.
JES3 Data	EPL-JES3-DATA	EPL3-	These fields pass JES3 statement information from JOB/SCAN to the standards program.
JES FORMAT PR	EPL2-JES-FORMAT-PR	EPL2P-	These fields pass JES3 /**FORMAT PR statement information from JOB/SCAN to the standards program.
Comment Card Data	EPL-COMMENT-CARD-DATA	N/A	This data element consists of a single 80-character field containing the entire contents of the COMMENT statement to be passed to the standards program.
Control Card Data	EPL-CONTROL-CARD	N/A	Table of control cards.
INCLUDE Data	EPL-INCLUDE-DATA	EPLI-	These fields pass the include member name contained in the INCLUDE statement to the standards program.
PROC Data	EPL-PROC-DATA	EPLP-	These fields pass PROC statement and EXEC PROC statement information from JOB/SCAN to the standards program.
IF THEN Data	EPL-IF-THEN-STATEMENT	EPLF-	These fields pass IF-THEN statement information from JOB/SCAN to the standards program.
XMIT Data	EPL-XMIT-DATA	EPLT-	These fields pass XMIT statement information from JOB/SCAN to the standards program.
EXECUTE Data	EPL-EXECUTE-DATA	EPLX-	These fields pass EXEC PGM= statement information from JOB/SCAN to the standards program.
DD Data	EPL-DD-STATEMENT	EPLD-	These fields pass DD information from

Data Element	EXIT PARM	Field-naming Prefix	Description
			JOB/SCAN to the standards program.
OUTPUT Data	EPL-OUTPUT-STATEMENT	EPL0-	These fields pass OUTPUT statement information from JOB/SCAN to the standards program.
Table Handling for User-defined Control Tables	EPL-USER-RCT, and EPL-USER-RCT-DATA	EPLU-	Two fields are associated with this data element. The first field is set within the standards program; it designates the type of table handling to be performed when reading the user-defined Control Tables in JSOPT02 and JSOPT03. Some functions are adding, searching, sorting, and deleting entries; a separate module within the load library then performs the function based on the request made. The second field is an 80-character work area used in passing the user-supplied Control Table entries to and from the standards program.
JOB/SCAN Errors Data	EPL-JOB-SCAN-ERRORS	EPLM-	Table contains JCL validation errors detected in the current JCL statement. Each table entry includes the error message number and severity level.
Control Statement Data	EPL-CONTROL-STATEMENTS	EPLC-	These fields pass control card information from JOB/SCAN to the standards program.
Operator Command Data	EPL-OPERATOR-COMMAND	N/A	The field of this CEPL section is used to pass a JCL command statement image from JOB/SCAN to the Standards program. This is not the COMMAND statement.
Data Element	EPL-ALIGN-CHECK	EPLAC-	This is to ensure that the correct copy

Data Element	EXIT PARM	Field-naming Prefix	Description
Integrity Check			of the CEPL is being used, per the current release of JOB/SCAN.

---

# Program Name Exit

The program name exit function provides JOB/SCAN the capability of recognizing a program name coded in an EXEC PGM statement, and calling another program (the program name exit). The program name exit determines the actual program to be executed by MVS, by evaluating data in the EXEC PGM statement PARM field or in the file indicated by a ddname in the Control Table CB entry. The program name exit can return an error message to JOB/SCAN to be printed on the Structured JCL Listing.

- [Setting Up the Program Name Exit](#)
- [Program Name Exit: How It Works](#)

## Setting Up the Program Name Exit

### *To set up the program name exit*

1. Modify the sample program name exit to perform the necessary functions.
2. Compile and link the program name exit program.
3. Code the Control Table entries and the Access Control Cards (ACC) Run Option.

### Modifying the Sample Program Name Exit

Modify the sample program name exit in the PROGNAME member of the SJJSSAMP library, following the instructions in [Program Name Exit: How It Works](#) for guidelines regarding inserting your code.

### Compiling and Link the Program

Compile and link the exit program into a library that is available to JOB/SCAN at run time. Compile the COBOL exit program using the ISPF edit macro CCEXIT found in SJJSSAMP.

### Coding Control Tables and ACC Run Option

Code a Control Table CB entry for each program name that requires calling a program name exit. Refer to [Control Table CB: Program Name Exit](#) for information about coding Control Table CB.

If the Control Table CB entry passes control cards by means of a ddname that points to a partitioned dataset and member, code a corresponding Control Table AB entry, and an Access Control Cards (ACC) Run Option specifying YES, in the appropriate SJJSPARM member. Refer to [Run Option Statements](#) for information about Run Options.

For batch execution, an Access Control Cards statement can be included in the Run Options section of the SYSIN stream.

## Program Name Exit: How It Works

JOB/SCAN and the program name exit communicate by passing information in predefined fields. JOB/SCAN makes calls to the program name exit to allow the exit to evaluate the information in those fields. The program name exit can check the actual program name to be executed by MVS, and can return an error message to be printed on the Structured JCL Listing.

## Calls to the Program Name Exit

Three types of calls are made to the program name exit to enable the exit to perform necessary functions at the appropriate time:

- Process call: JOB/SCAN calls the program name exit with a process call when it encounters an EXEC PGM statement with a corresponding Control Table CB entry. If the Control Table CB entry contains a ddname, JOB/SCAN calls the program name exit with a process call once for each record in the file identified by the ddname.
- EOD (end-of-data) call: One call for each EXEC PGM statement processed. The EOD call is issued after the last process call has been issued for that statement or its associated data file.
- Final call: One last call made to the program name exit at the end of the JOB/SCAN run. The exit can perform any necessary housekeeping tasks. A sample program name exit, PROGNAMEX, is included in the SJJSSAMP library. This sample exit provides the structure to handle all three call types and can be modified to perform the necessary functions. Passing the Data Items

These fields, used for communication between JOB/SCAN and the Program Name Exit, are included in the linkage section of the exit. The below figure shows sample code required for a program name exit.

### Communication Fields for Program Name Exit

```

*****
LINKAGE SECTION.
*****
01 PROGRAM-NAME-EXIT.
   05 PNE-CALL-DATA.
      10 PNE-CALL-TYPE          PIC X(01) .
         88 PNE-CALL-TYPE-PROCESS VALUE 'P' .
         88 PNE-CALL-TYPE-EOD     VALUE 'E' .
         88 PNE-CALL-TYPE-FINAL   VALUE 'F' .
      10 PNE-PRODUCT-TYPE      PIC X(01) .
         88 PNE-PRODUCT-DOCUTEXT  VALUE 'D' .
         88 PNE-PRODUCT-INFO-X    VALUE 'I' .
         88 PNE-PRODUCT-JOBSCAN   VALUE 'J' .
      10 PNE-PROGRAM-NAME      PIC X(08) .
      10 PNE-PROGRAM-PARM      PIC X(100) .
      10 PNE-DDNAME            PIC X(08) .
      10 PNE-CONTROL-CARD      PIC X(80) .
      10 FILLER                PIC X(70) .
   05 PNE-RETURN-DATA.
      10 PNE-NEW-PROGRAM-NAME   PIC X(08) .
         88 PNE-FINISHED         VALUE LOW-VALUES .
      10 PNE-ERROR-SEVERITY-CODE PIC S9(04) COMP .
      10 PNE-ERROR-MESSAGE      PIC X(52) .
      10 FILLER                PIC X(50) .
EJECT

```

This table lists the field used by the sample, and describes PROGRAM-NAME-EXIT:



OFFSETS				
DEC	HEX	TYPE	LEN	DESCRIPTION
				CALL TYPE (INPUT)
00	00	CHAR	1	<p>Three different reasons why the exit is called:</p> <p>'P' = PROCESS</p> <p>Program name coded on EXEC PGM will be in PROGRAM NAME FIELD. When a DDNAME has been coded, a PROCESS CALL TYPE will be made for every control card coded. When a DDNAME has not been coded, one PROCESS CALL TYPE will be made for only the EXEC PGM statement.</p> <p>'E' = END OF DATA</p> <p>When a DDNAME has been coded, an END OF DATA call will be made after all the control cards have been passed to the PROGRAM NAME EXIT. 'F' = FINAL CALL FOR EXIT</p> <p>A FINAL CALL is made to let the PROGRAM NAME EXIT know that this is the last time the exit will be called. This provides an opportunity for the exit to perform any cleanup.</p>
01	01	CHAR	1	<p>PRODUCT TYPE (INPUT)</p> <p>'J' = JOB/SCAN</p>
02	02	CHAR	8	<p>PROGRAM NAME (INPUT)</p> <p>The PROGRAM NAME coded on the EXEC PGM statement.</p>
				PROGRAM PARM (INPUT)
10	0A	CHAR	100	<p>The PARM that MVS would pass to the program being called. This may not always be the PARM field coded on the EXEC PGM statement due to PARM overrides when coded in a PROC.</p>
				DDNAME (INPUT)
110	6E	CHAR	8	<p>The DDNAME coded on the DD statement and in a Control Table CB entry (optional).</p>
				CONTROL CARD (INPUT)
118	76	CHAR	80	<p>The CONTROL CARD coded either in a PDS member or SYSIN dataset if DDNAME is coded in a Control Table CB entry.</p>
				NEW PROGRAM NAME (OUTPUT)
268	10C	CHAR	8	<p>This field is not used for JOB/SCAN.</p>

## OFFSETS

DEC	HEX	TYPE	LEN	DESCRIPTION
				ERROR SEVERITY CODE (OUTPUT) This field contains the user-specified severity code:
276	114	FIXED	2	H'0' - Advisory Message H'4' - Warning Message H'8' - Error Message H'20' - Catastrophic Message
				ERROR MESSAGE (OUTPUT)
278	116	CHAR	52	This field contains a 3-byte message number followed by a message to be placed in the Structured JCL Listing.
				MEMBER NAME
330	14A	CHAR	8	The name of the Program Exit being called.
338	152	CHAR	8	This is an optional parameter that is passed to all Exit Programs.

## User Abends Issued by JOB/SCAN

This table lists user abend codes, the related modules, and explanations.

Abend Code	Load Module	Csect	Explanation
0001	J00YCOM	D00YDQ	An uncorrectable I/O error has occurred while processing a dataset, and a reason code other than 00 has been returned from the SYNADRLS macro processing. See message DSS0611 in the output error report for a description of the I/O error.
0002	J00YCOM	D00YOIC	An uncorrectable I/O error has occurred while processing a dataset, and a reason code other than 00 has been returned from the SYNADRLS macro processing. See message DSS0711 in the output error report for a description of the I/O error.
0097	J00XLP	D00XLP	JOB/SCAN has encountered an internal subroutine error. Call ASG Customer Support for assistance.
0951	J00XPJ	D00XPJ	Invalid Processing Option specified. Internal diagnostic error. Contact ASG Customer Support for assistance.
0952	J00XPJ	D00XPJ	Invalid return from ICAT update. Internal diagnostic error. Contact ASG Customer Support for assistance.
0991	J00YOPC2	D00YOPC2	Unable to access the TWS database.
U4001	J06YREX2	D06YREX2	Caller did not supply an input parameter list. This module requires one input parameter; the address of the Exit Parameter List (EPL).
U4002	J06YREX2	D06YREX2	Invalid Exit Parameter List. EPL field C2LITERL did not contain the character string C'DSSI EPL END'.
U4003	J06YREFP	D06YREGT	Address of User Area in REXX environment block is zero. REXX Function JSGET.
U4004	J06YREFP	D06YREGT	Contents of User Area eye-catcher has changed. REXX Function JSGET.
U4005	J06YREFP	D06YREPT	Address of User Area in REXX environment block is zero. REXX Function JSPUT.
U4006	J06YREFP	D06YREPT	Contents of User Area eye-catcher has changed. REXX Function

Abend Code	Load Module	Csect	Explanation
			JSPUT.
U4007	J06YREFP	D06YREMG	Address of User Area in REXX environment block is zero. REXX Function JSMSG.
U4008	J06YREFP	D06YREMG	Contents of User Area eye-catcher has changed. REXX Function JSMSG.
U4009	J06YREFP	D06YREDP	Address of User Area in REXX environment block is zero. REXX Function JSDUMP.
U4010	J06YREFP	D06YREDP	Contents of User Area eye-catcher has changed. REXX Function JSDUMP.
U4011	J06YREFP	D06YRETB	Address of User Area in REXX environment block is zero. REXX Function JSTABLE.
U4012	J06YREFP	D06YRETB	Contents of User Area eye-catcher has changed. REXX Function JSTABLE.
U4013	J06YREX3	D06YREX3	Caller did not supply an input parameter list. This module requires one input parameter; the address of the Exit Parameter List (EPL).
U4014	J06YREX3	D06YREX3	Invalid Exit Parameter List. EPL field C2LITERL did not contain the character string C'DSSI EPL END'.

# Selection Exit Interface

Selection Exits pre-process the JCL. This enables the user to change/add/delete JCL statements which the Exit passes to JOB/SCAN. Before JOB/SCAN validates the JCL, the lines of JCL are passed one at a time to the Selection Exit program. The Exit evaluates the line, makes changes if necessary, and requests an action based on the evaluation. This facility gives JOB/SCAN the ability to duplicate the functions of JES exits and other software that change the JCL at submission-time.

- [Overview](#)
- [Selection Exit Defined In Edit Macros](#)
- [INSJOB: Selection Exit to Insert a Jobcard](#)
- [JOBUSER: Selection Exit to Insert 'USER=' Parameter](#)

## Overview

Selection exits can be written in any LE-compliant language. These exits are passed 80-byte card images, and issue a return code telling JOB/SCAN what action to take with a JCL statement:

- Use as is
- Use as modified
- Do not include this JCL in this run
- Add this statement and call the Selection Exit again with the same input statement

For a sample Selection Exit, refer to the CA7J1 member in the SJJSSAMP library.

### Limitations

JOB/SCAN passes only the EXECUTION JCL to the Selection Exit. Any INCLUDE statements or called Procedures are not passed to the Selection Exit.

JSCAN is the Edit macro that displays inserted the lines from a selection exit; JEM does not.

### Process Overview

#### *To set up the selection exit*

1. Modify the sample Selection Exit to perform the necessary functions.

Modify the sample Selection Exit in the CA7J1 member of the SJJSSAMP library according to the instructions in [Modifying the Sample Selection Exit](#) next. CA7J1 performs special processing for CA-7 #J cards. Refer to [Handling #J Cards with the CA-7 Scheduler Interface](#), for information on handling #J cards.

2. Compile and link the exit program.

Compile and link the exit program to a library that is available to JOB/SCAN during run time. Use the ISPF edit macro CCEXIT found in SJJSSAMP to compile and link edit the exit program.

3. Enable the interface to JOB/SCAN.

For foreground or panel-assisted batch execution, type the Selection Exit name (and parameter, if any) on the JCL Validation panel.

For batch execution without panels, include a Selection Exit statement in the Input Selection section of the SYSIN stream before the first member name selected. For example:

```
//SYSIN DD *
INPUT SELECTION
      SELECTION EXIT=SELPGM,' PARM1'
      SELECT      =JOB1
DSN      =
```

The Selection Exit is now automatically called for each line of JCL before JOB/SCAN processes it.

## JCL Validation Panel

### To prepare a selection exit

- Type the Selection Exit name (and parameter, if any) in the Selection Exit (and Parm fields) of the JCL Validation panel.

### Enabling a Selection Exit

```
----- JOB/SCAN - JCL VALIDATION -----
COMMAND ==>

      F - Foreground execution mode          P - Processing Options
      B - Batch execution mode              E - Edit JCL member
                                          BR - Browse JCL member

Input Library  ==> SYSUSR.JCL
Input Type    ==> J          (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ==> PDS       (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> LEDGRO   (Single member if BATCH with TRANSMIT)
Exclude Members ==>        (Not valid for BATCH with TRANSMIT)
First Proclib ==>
Proclib Type  ==>          (PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
Report Heading ==>
Display        ==> Y (Yes/No/ErrorsOnly)  Browse/View ==> B (B/V)
Hardcopy       ==> N (Yes/No/ErrorsOnly/SpaceOnly/JobErrors)
Options in Effect Listing ==> N (Y/N)      SJL PDSOUT ==> N (Y/N)

Selection Exit ==> SELEXIT      Parm ==> PARM1233
```

## Edit Macro Processing

### To use a selection exit

- With a JOB/SCAN Edit macro, type the SELEXIT parameter with the exit name in parentheses.

### Original JCL

```
EDIT ---- DSSI.PROD.JCLLIB(IEFBR14) - 01.02 ----- COLUMNS 001 072
COMMAND ==> JSCAN SELEXIT(INSJOB) SCROLL ==> PAGE
***** ***** TOP OF DATA *****
000100 //STEP1 EXEC PGM=IEFBR14
000200 //DD1 DD DSN=ASGUSR.WORK.CNTL,
000210 // DISP=(,CATLG)
***** ***** BOTTOM OF DATA *****
```

## Resulting SJL

```

***** TOP OF DATA *****
JOB/SCAN          STRUCTURED JCL LISTING
-----
//INSERTED JOB (A), 'A', CLASS=X
//STEP1 EXEC PGM=IEFBR14                      000100
//DD1          DD DSN=ASGUSR.WORK.CNTL,        000200
//              DISP=(,CATLG)                 000210
***ERROR - DSS5130E - FILE IS ALREADY CATALOGED
*ADVISORY - DSS8900A - DSN = "ASGUSR.WORK.CNTL "
*** ERROR SUMMARY AND COUNTS ***
  1 ADVISORY LEVEL
  0 WARNING LEVEL
  1 ERROR LEVEL
  2 TOTAL ISSUED          0 SUPPRESSED.
  THE HIGHEST SEVERITY CODE ENCOUNTERED WAS 08.
  THE SELECT EXIT CALLED FOR THIS RUN WAS "INSJOB".
***** BOTTOM OF DATA *****

```

## Edit Macro Processing With A Parameter

### To include a parameter for a selection exit with a JOB/SCAN Edit macro

- Type the SELEXIT parameter with the exit name in parentheses and use the SELPARM (SP) parameter. (CA7J1 is located in the SJJSSAMP library).

### Example showing use of SELPARM macro parameter

```

EDIT ---- DSSI.JOBSCAN.SJJSSAMP(CA7JCL) - 01.00 ----- COLUMNS 001 072
COMMAND ==> JSCAN SELEXIT(CA7J1) SELPARM(001)                SCROLL ==> PAGE
***** ***** TOP OF DATA *****
000001 //JOB/JCL JOB ,CLASS=A,MSGCLASS=X,NOTIFY=ASGUSR
000002 //STEP1 EXEC PGM=IEFBR14
000003 //SYSPRINT DD SYSOUT=X
000004 #JI, ID=002
000005 //DD1          DD DSN=ASGUSR.WORK.CNTL, DISP=SHR
000006 #JEND
000007 #JI, ID=001
000008 //DD2          DD DSN=ASGUSR.WORK.XXXX, DISP=SHR
000009 #JEND
***** ***** BOTTOM OF DATA *****

```

## Passing The Data Items

These fields, used for communication between JOB/SCAN and the Selection Exit, are included in the linkage section of the exit:

```

LINKAGE SECTION.
01  JCL-STATEMENT          PIC X(80) .
01  USER-PARM             PIC X(34) .
01  CALL-SWITCH           PIC X .
88  FIRST-CALL-FOR-JOB    VALUE 'Y' .
88  CALL-SWITCH-END-OF-ALL-MEMBERS VALUE 'Z' .
01  MEMBER-NAME           PIC X(10) .
01  USER-EXIT-WORD       PIC X(04) .

```

where:

**JCL-STATEMENT** contains the 80-character JCL statement. Only the statements from the input JCL library are passed to the exit; cataloged PROC statements are not passed.

**USER-PARM** is an optional user parameter that can be passed to the exit program in the USER-PARM field when the Selection Exit name is specified to JOB/SCAN.

**CALL-SWITCH** is set to Y for the first call to the exit for each member. It is set to Z when all members have been processed. The exit program must reset this switch unless all members have been processed (CALL-SWITCH=Z).

**MEMBER-NAME** contains the name of the member in which the current JCL statement resides.

**USER-EXIT-WORD** is a word (4 bytes) which is for the user exits. This word is passed to all exits.

### Writing A Selection Exit

#### *To write a Selection Exit*

1. Determine what to evaluate on any JCL statement.
2. Code the logic necessary to perform the evaluation and any action that needs to be taken.

For example, turn the line into a comment by placing a `//*` in front of it.

3. Set the return code to the appropriate value to specify the action for JOB/SCAN to take (see [Selection Exit Return Codes](#)).
4. Compile and link the selection exit using the edit macro CCEXIT in SJJSSAMP.
5. Turn on the Selection Exit interface to JOB/SCAN.

### Modifying the Sample Selection Exit

If you choose to write a Selection Exit, copy member CA7J1 from the SJJSSAMP library. Modify it to perform the necessary functions. See [Selection Exit Return Codes](#) for details on return codes to set.

### Selection Exit Return Codes

The Selection Exit program generates return codes that affect the way JOB/SCAN processes input statements, as shown in the this table.

Return Code	Action JOB/SCAN takes
0	Process the line as is. The Selection Exit program has not changed this statement.
4	Process the line as edited. The Selection Exit program has changed this statement.
8	Do not process this line.
12	Insert this new statement and call the Selection Exit again using the original statement (used to add data to the input stream).
16	Stop processing this member.
20	Terminate the JOB/SCAN run.

Use return codes to specify the action JOB/SCAN should take. For example, to insert a line, place the line in the JCL-STATEMENT field and set the return code to 12.



## Selection Exit Defined In Edit Macros

In some cases users do not want to pass a parm when using the Edit Macros. When these situations occur, the customer can modify their Edit Macros (JEM or JSCAN) to handle these types of situations:

- JEM1 Use the CA-7 Selection Exit for schedule ID 001.
- JEM2 Use the CA-7 Selection Exit for schedule ID 002.
- JEM3 Use the CA-7 Selection Exit for schedule ID 003.
- JEM4 Use the CA-7 Selection Exit for schedule ID 004.

### Selection Exit Modifications

```

EDIT ---- DSSI.PROD.SJJSCLST(JEM1) - 01.00 ----- COLUMNS 001 072
COMMAND ==> SCROLL ==> PAGE
***** ***** TOP OF DATA *****
000001      PROC 0  DJSNAME(DSSI.JOBSCAN)
000002      /* (C) 1988-2017 ASG TECHNOLOGIES GROUP, INC.
000003      /* ALL RIGHTS RESERVED.
000004      CONTROL END(ENDO) NOMSG NOFLUSH
000005      /*CONTROL END(ENDO) MSG LIST CONLIST SYMLIST NOFLUSH
000006      ISREDIT MACRO (PARMSTR)
000007
000008      IF &DJSNAME = &STR(XXXXX.XXXXX) THEN +
000009      DO
000010          ISPEXEC SETMSG MSG(JM022R)
000011          GOTO ENDING
000012      ENDO
000013
000014      SET &RCLALC = 'Y'
000015      ISPEXEC SELECT PGM(J00YCKAL) PARM(JLRCL)
000016      IF &LASTCC NE 0 THEN +
000017      DO
000018          SET &RCLALC = 'N'
000019          ALLOC DDN(JLRCL) DSN('&DJSNAME..SJJS Parm') SHR
000020      ENDO
000021
000022      ISPEXEC SELECT PGM(J0AYEMU) PARM(PR(E) &PARMSTR SE(CA7J1) SP(001) A
000023
000024      ENDING: +
000025      IF &RCLALC = 'N' THEN FREE DDN(JLRCL)
000026      EXIT

```

## INSJOB: Selection Exit to Insert a Jobcard

The INSJOB member in the SJSSAMP library contains the source code for a Selection Exit to insert a JOB card.

The below figure shows original JCL without a JOB card.

### Original JCL before INSJOB exit

```

EDIT ---- DSSI.PROD.CNTL(CA7JCL) - 01.00 ----- COLUMNS 001 072
COMMAND ==> JSCAN SELEXIT(IINSJOB) SCROLL ==> PAGE
***** ***** TOP OF DATA *****
000001 //*
000002 //STEP1 EXEC PGM=IEFBR14
000003 //SYSPRINT DD SYSOUT=X
000004 //DD1 DD DSN=ASGUSR.WORK.CNTL,DISP=SHR
***** ***** BOTTOM OF DATA *****

```

The below figure shows the modified JCL with an added JOB card.

#### Modified JCL after INSJOB exit

```

EDIT ---- DSSI.PROD.CNTL(CA7JCL) - 01.00 ----- COLUMNS 001 072
COMMAND ==>                                SCROLL ==> PAGE
***** ***** TOP OF DATA *****
000001 //JOBJCL JOB ,CLASS=A,MSGCLASS=X,NOTIFY=ASGUSR
000002 //*
000003 //STEP1 EXEC PGM=IEFBR14
000004 //SYSPRINT DD SYSOUT=X
000005 //DD1      DD DSN=ASGUSR.WORK.CNTL,DISP=SHR
***** ***** BOTTOM OF DATA *****

```

## JOBUSER: Selection Exit to Insert 'USER=' Parameter

The JOBUSER member in the SJSSAMP library contains the source code for a Selection Exit to insert a parameter.

The below figure shows the JCL to insert USER= in a JOB card.

#### JCL before JOB/SCAN runs a User exit to add code

```

EDIT ---- DSSI.PROD.CNTL(CA7JCL) - 01.00 ----- COLUMNS 001 072
COMMAND ==> JSCAN SELEXIT(JOBUSER)                SCROLL ==> PAGE
***** ***** TOP OF DATA *****
000001 //JOBJCL JOB ,CLASS=A,MSGCLASS=X,NOTIFY=ASGUSR
000002 //STEP1 EXEC PGM=IEFBR14
000003 //SYSPRINT DD SYSOUT=X
***** ***** BOTTOM OF DATA *****

```

The below figure shows the JCL with USER=JO0001 added by the Exit.

#### JCL after JOB/SCAN runs a User exit to add code

```

EDIT ---- DSSI.PROD.CNTL(CA7JCL) - 01.00 ----- COLUMNS 001 072
COMMAND ==>                                SCROLL ==> PAGE
***** ***** TOP OF DATA *****
000001 //JOBJCL JOB ,CLASS=A,MSGCLASS=X,NOTIFY=ASGUSR,
000002 //USER=JO0001,                          * GENERATED BY JOB/SCAN *
000003 //STEP1 EXEC PGM=IEFBR14
000004 //SYSPRINT DD SYSOUT=X
***** ***** BOTTOM OF DATA *****

```

# Standards Information

This documentation is designed for experienced JOB/SCAN users who are tasked with designing and implementing JCL standards. It can be used by non-programmers. Using this documentation does not make you a COBOL nor a REXX programmer, but tells you what you need to know to implement standards.

Topic	Description
<a href="#">Standards Overview</a>	Explains the concepts of writing a standard.
<a href="#">Using the Standards Facility</a>	Explains the basics of using standards in the three JOB/SCAN operating modes.
<a href="#">Standards Facility Error Messages</a>	Describes how to create, use and maintain error message in the Standards facility.
<a href="#">Using Extended Parameter List Fields and Tables</a>	Describes the EPL and the tables it comprises.
<a href="#">COBOL Standards Implementation Using the Prompter</a>	Describes the Standards Utility Menu, how to set up the Prompter, and the Standards Prompter, including easy-to-follow steps for adding, replacing, and deleting standards.
<a href="#">COBOL Primer</a>	Introduces the COBOL language, which is used to write a standards program.
<a href="#">COBOL Standards Testing and Debugging</a>	Provides useful tips and techniques for testing standards programs.
<a href="#">REXX Standards</a>	An overview of using REXX to write standards.
<a href="#">REXX Standards Implementation</a>	Describes the use of REXX in writing standards for JOB/SCAN.
<a href="#">REXX Programming Notes</a>	Describes major REXX language components.
<a href="#">REXX Standards Errors and Abends</a>	Explains REXX abends.
<a href="#">REXX Functions</a>	Describes the REXX functions used by the standards facility.
<a href="#">SJSSTDL Library Members</a>	Provides information on how to use SJSSTDL library members and explains about the SJSSTDL library members.
<a href="#">Call Types</a>	Describes the Standards Facility processes call types.
<a href="#">Common Questions about Writing Standards</a>	Answers questions frequently asked about writing standards for JOB/SCAN.

# Standards Overview

This topic explains the concepts of writing a standard.

- [Introduction to Standards](#)
- [The Standards Process](#)

## Introduction to Standards

The acceptability of JCL goes beyond mere syntax and run-time correctness. Most sites require that JCL meet certain conditions. For instance: is a mandated parameter present and does it have an acceptable value? Do dataset names follow established conventions? Is a PROC or program executed from the correct library? These site-defined requirements cannot be tested by syntax and run-time checkers; they must be tested by other means.

The JOB/SCAN Standards Facility (herein called the Standards Facility) integrates a site's additional requirements into JOB/SCAN's comprehensive JCL validation.

The Standards Facility helps implement and enforce site-specific requirements in programs called *standards* (historically called standards exits or standards routines). The statements that implement these requirements are often written in COBOL, though other languages can be used. (See [REXX Standards](#) through [REXX Standards Errors and Abends](#) for guidance about writing standards in REXX.)

Standards help protect and manage systems assets. Standards with JOB/SCAN can:

- Enforce job and dataset naming conventions.
- Enforce space allocation restrictions.
- Monitor device type and job class usage.
- Search recently acquired software for certain device types or SYSOUT classes.
- Preview the impact of proposed changes to site requirements.
- Detect violations and post error messages to the Structured JCL Listing (SJL).

### How to Use the Standards Facility

#### *To use the Standards Facility*

1. Design and write standards statements using appropriate IF/THEN statements to test JCL for specific requirements.

There are two different tools that can be used to write standards:

- The JOB/SCAN Standards Prompter panel.
  - Any text editor (the only way for REXX standards).
2. Insert the statements into a standards program.
  3. Design messages using these criteria:

- Statement types - the JCL statement to be examined.
  - Message text - a message containing up to 48 characters describing the standards violation.
  - Message number - a three-digit number from 000 to 999 identifying the standards violation; each message number should be unique.
  - Severity code - a number usually from 0 to 19 indicating the severity of the standards violation.
4. Insert the messages into the code or Standards Message Facility.
  5. Compile the standards and link it into a load library. Optional for REXX standards.
  6. Debug the standards in a test environment.
  7. Promote the standards to production.
  8. Tell JOB/SCAN users to use the standards. See [Using the Standards Facility](#).

With a standards program in effect, site-specific requirements violations are flagged with the associated error message directly in the SJL. JOB/SCAN applies standards by:

- Processing each logical JCL statement fully for syntax and run-time validation.
- Passing the information for each logical JCL statement to standards.

Using the Standards Prompter makes the job easier by automatically generating most of the COBOL for standards and the JCL to compile it. Using a text editor, for instance, ISPF Edit, allows more power and flexibility; this approach requires more knowledge of COBOL.

### Important Features of the Standards Facility

This table describes the features available with the Standards Facility.

Feature	Benefit
Easy access to values from the JCL	Values from every possible JCL keyword are extracted automatically and are readily available. See <a href="#">Using Extended Parameter List Fields and Tables</a> .
Accommodation of run-time values	Because different processing situations may require specific run-time information, it is possible to pass a run-time parameter to the standards program.
Extended Messages	Effective standards implementation requires that information for correcting violations be readily available to the user. With JOB/SCAN, extended message text can be inserted into the SJL at the point of violation, to relate the error immediately to the explanation. See <a href="#">Standards Facility Error Messages</a> .
Ease of implementation	A kit included with JOB/SCAN helps define and implement standards easily.
Run-time selection of standards	JOB/SCAN provides the flexibility to select the set of standards applicable to each run.
Forced use of standards	JOB/SCAN can be set up to run with a predetermined set of standards in every execution.



JCL standards issues differ from reformatting issues in that standards deal with how the JCL is coded, not what the statements look like or how they are aligned or arranged.

### Statement Types

These are the statement types for which standards may be written:

Comment	EXEC PGM	JES2	PROC and EXEC PROC
Control cards	IF-THEN	JES3	SET
DD	INCLUDE	JOB	Unknown
ELSE	JCLLIB	OUTPUT	XMIT

## The Standards Process

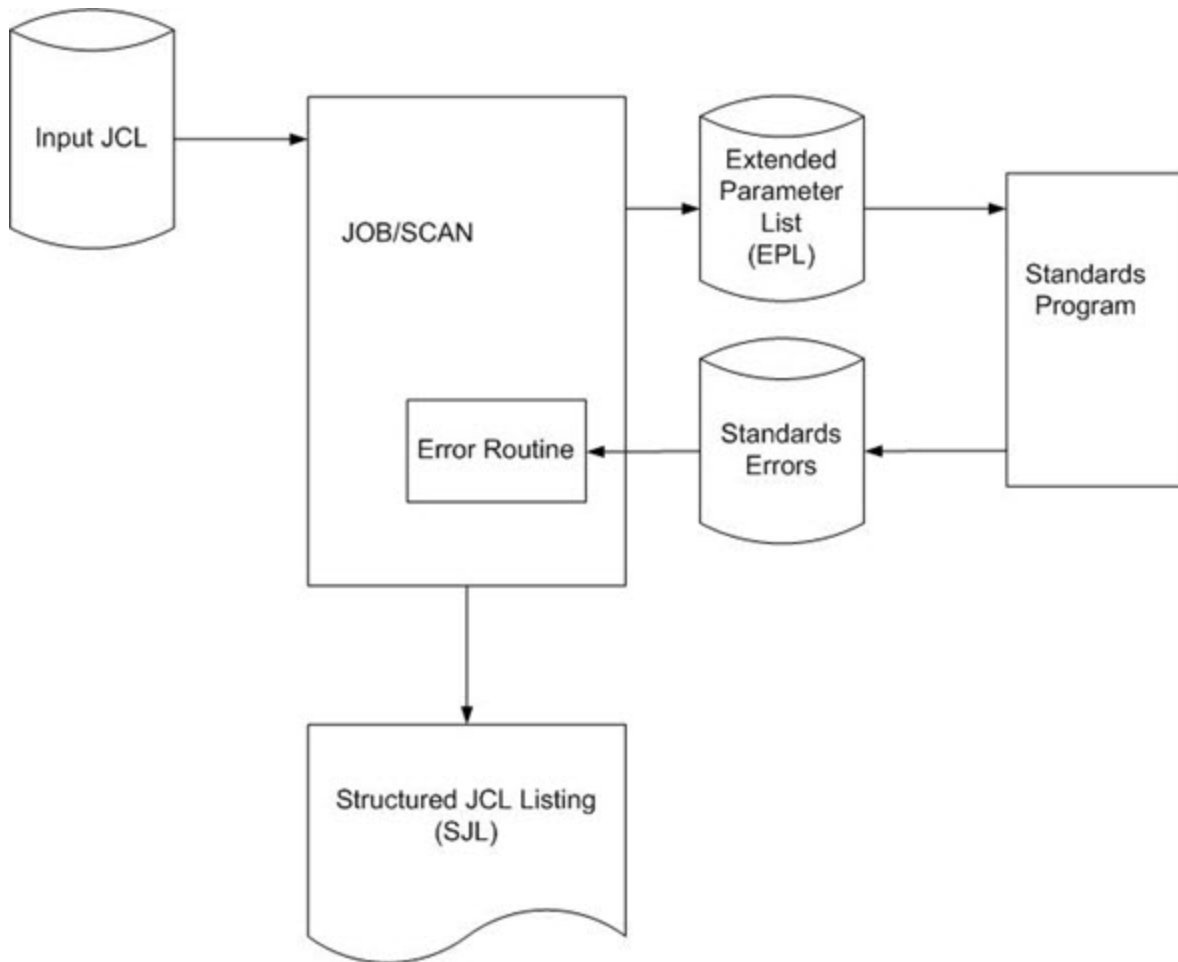
With standards processing turned on, as it processes each JCL statement in a validation run, JOB/SCAN:

1. Places each parameter value found in the various statements into a special area called Extended Parameter List (EPL).
2. Calls standards, which can examine the contents of the JCL statement in question. The Standards place error messages into a special message area.
3. Picks up the messages.
4. Places them in the Structured JCL Listing (SJL).
5. Processes each remaining JCL statement through steps 1 - 6.



There is no difference in this process for compiled standards (written in COBOL or Assembler) or REXX standards.

## JOB/SCAN Standards Architecture



## Using the Standards Facility

The Standards Facility is available from each of the three JOB/SCAN operating modes: ISPF panels, Edit macros, and batch.

- [Using Standards in ISPF Mode](#)
- [Using Standards with Edit Macros](#)
- [Using Standards in Batch Mode](#)

### Using Standards in ISPF Mode

Before accessing a standards program from the JCL Validation panel, make sure its module is in a library that is allocated either to ISPLLIB, STEPLIB, or JLLD. Alternatively, it can be in a LINKLST library. In the case of REXX standards, see [REXX Standards Implementation](#) for information about appropriate libraries.

#### *To validate JCL using site standards*

1. Select Validate JCL from the ASG-JOB/SCAN Primary Menu panel.
2. Follow the procedures for entering values in the fields of the JCL Validation panel. (Standards can also be used through the 'one-pass' panel.)
3. Type the name of the standards program in the Standards Name field.
4. Optionally, type a parameter in the Parm field. JOB/SCAN can pass up to 32 characters of information entered in the Parm field.
5. Press Enter.

The site standards coded in the standards program will be validated automatically, along with the validation done by JOB/SCAN.



If the execution panel does not display the Standards Name and Parm fields, access is not available to the Standards Facility. Ask the JOB/SCAN administrator for access.

In some cases, a JCL standards member name automatically displays in the Standards Name field of the JCL Validation panel. If this value cannot be over-typed, it means that the JCL standards program shown is required and cannot be changed. The Product Administrator can change this requirement using Access Control statements in JSOPT01.

#### **Standards Parameter**

Because different processing situations might require specific run-time information, it might be necessary to pass a run-time parameter to the standards program. If the standards program is set up to accept a parameter, specify it on the panel.



## Specifying a Standards Program

```

----- JOB/SCAN - JCL VALIDATION -----
COMMAND ==>

      F - Foreground execution mode          P - Processing Options
      B - Batch execution mode              E - Edit JCL member
                                          BR - Browse JCL member

Input Library  ==> ASGUSR.JCL
Input Type    ==> J          (J - JOB JCL, P - PROC, or L - Selection List)
Library Type  ==> PDS       (PDS, PAN, LIB, LIB-3, or SEQ)
Select Members ==> LEDGR0   (Single member if BATCH with TRANSMIT)
Exclude Members ==>         (Not valid for BATCH with TRANSMIT)
First Proclib ==>
Proclib Type  ==>         (PDS, PAN, LIB, or LIB-3)

Structured JCL Listing:
Report Heading ==>
Display        ==> Y (Yes/No/ErrorsOnly)  Browse/View ==> B (B/V)
Hardcopy       ==> N (Yes/No/ErrorsOnly/SpaceOnly/JobErrors)
Options in Effect Listing ==> N (Y/N)      SJL PDSOUT ==> N (Y/N)

Selection Exit ==>          Parm ==>
Standards Exit ==> SYSEXIT  Parm ==> PARM12

```

## Using Standards with Edit Macros

When using the JSCAN or JEM Edit macros, standards program can be specified by typing the optional parameter STDPGM (progname) on the command line.

### Standards Parameter

If a user parameter is required, type the optional parameter UPARM on the command line as well. See [Validating JCL with Edit Macros](#) for more about the JOB/SCAN Edit macros and parameters.

## Using Standards in Batch Mode

To execute JOB/SCAN with a standards program in batch mode, a STANDARDS NAME Run Option statement must be coded. This statement contains the name of the standards program and up to 32 characters of parameter information (enclosed in single quotes).

The STANDARDS NAME Run Option can be coded in JSOPT02, JSOPT03, or the JOB/SCAN SYSIN stream. If placed in the SYSIN stream, STANDARDS NAME must follow a Run Option group-level statement. If the standards program is not in a JOBLIB or STEPLIB allocation and is not link-listed, include a JLLOD DD statement in the JCL. For example:

```
//JLLOD DD DSN=loadlib,DISP=SHR
```

This SYSIN stream shows how to set up the JCL to execute JOB/SCAN using the standards program CSMP. (This program is provided as a sample with JOB/SCAN.)

```

//JSMVS JOB 9999,'JOB/SCAN',CLASS=A
//JSSTEP01 EXEC JS6PROC,PRM=S
//SYSIN DD *
INPUT SELECTION
        DSN=DSSI.PROD.SJJSDEMO
        SELECT=PJGL0020

```

```
REPORT OPTIONS
  OPTIONS IN EFFECT=SUMMARY,PRINT
  STRUCTURED JCL LISTING=PRINT
RUN OPTIONS
STANDARDS NAME=CSMP,'parameter'
  REPORT HEADING='Test for Exit Processing'
```

---

## Standards Parameter

Specify the standards parameter on the STANDARDS NAME Run option:

```
STANDARDS NAME=CSMP,'parameter'
```

# Standards Facility Error Messages

A standards program can generate multiple error messages for each JCL statement validated. As described in this topic, specify a three-digit numeric value for the message, the text that makes up the short message, and the severity level of the message. Also specify a flag that determines whether to include a longer extended text from SJJSPARM library member DSSMSG90 for this error. This topic primarily discusses Standards Facility error messages from the COBOL language perspective. See [REXX Functions](#) for complete details about how to use error messages with REXX.

- [Standards Error Messages](#)
- [Default Error Message Handling](#)
- [Extended Error Messages](#)
- [Individual Control Over Standards Messages](#)
- [Changing or Suppressing JOB/SCAN Error Messages](#)

## Standards Error Messages

Error messages have the same format as other JOB/SCAN messages. The user message for Standards comprises two mandatory parts: the [Short Message Text](#) and the [Message Identification](#). An optional third part is the extended message; see [Extended Error Messages](#) for information about using extended text.

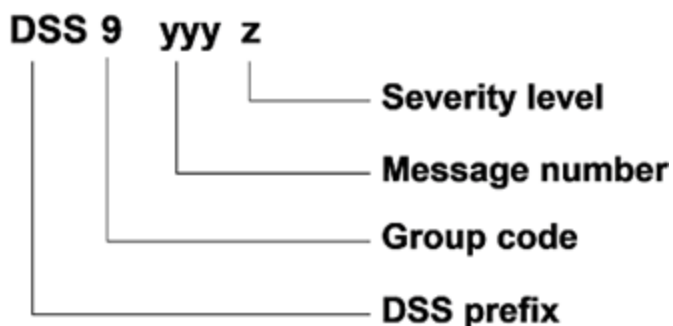
### Short Message Text

Message text up to 48 characters that briefly describes the standards violation. The text is used to build a message that will, like JOB/SCAN's syntax and run-time error messages, be displayed below the statement found to be in violation when JOB/SCAN analyzes JCL. If extended text is needed, see [Extended Error Messages](#) for information about using extended text. For example:

```
DSS9001E CLASS=A IS INVALID FOR PRODUCTION.
```

### Message Identification

This figure shows the format of the error message identification number. JOB/SCAN writes this number to the SJL and is used to identify extended message text.



### Severity Level

This table describes the severity level and return codes for error messages.

Severity Level	Return Codes	Meaning
A	0	Advisory message
W	1-4	Warning
E	5-19	Error message
C	20 and above	Catastrophic error; processing suspended

### Message Number

A three-digit message number from 000 to 999 to identify particular standards violations. Use a unique number for each message. Pad one- and two-digit number with zeros to make a three-digit number. ASG recommends establishing a numbering system for your messages. For example, all JOB statement standards could have message numbers from 100 to 199, while DD statements have message numbers from 200 to 299. If no value is coded, JOB/SCAN uses 999 when displaying the message. This value is assigned by JOB/SCAN and cannot be changed in the standards program.

### DSS Prefix and Group Code

Standards error messages always have a group code of 9. JOB/SCAN combines this number with 'DSS', the message number and the severity level. For example:

```
MOVE '608 CLASS PARAMETER NOT RIGHT'
```

JOB/SCAN builds the message number DSS9608 for the SJL appending A, W, E, or C depending on the severity assigned to it.

### SJL showing the extended text

```

STRUCTURED JCL LISTING      ASG, INC
DSNAME - CS082B.JCL      MEMBER - JSMVS
-----
|JOB - CS082BX                      MSGCLASS - X
|                                  CLASS - F
|PROGRAMMER NAME -NAME             ACCTG - ACCT
-----
| JOB JCL      | PROCLIB | SORT/UTILITY CARDS
-----
//CS082BX JOB ACCT, 'NAME', NOTIFY=CS082B,
//              MSGLEVEL=1, MSGCLASS=X, CLASS=F,
//              REGION=0M, TIME=1440
**WARNING - DSS9608W - CLASS PARAMETER NOT RIGHT
-----
This JOB class is not right. Set it to "P" and revalidate.
-----
...

```

## Default Error Message Handling

Using the ERRORTXT parameter in the Structured JCL Listing Report Option, the Product Administrator establishes the default error message style JOB/SCAN inserts in the SJL: either the short message or extended message:

```
SJL =ERRORTXT=EXTENDED
```

See [Reference Information](#) for more about the Structured JCL Listing Report Option.

The site default can be overridden by the standards program on a message-by-message basis. To display only the short message, use:

```
MOVE 'S' TO EPLC-ERRTEXT-LVL
```

To display the short and extended message text, use:

```
MOVE 'X' TO EPLC-ERRTEXT-LVL
```

If EPLC-ERRTEXT-LVL is left blank, the Structured JCL Listing Report Option default value determines the error text level.

## Extended Error Messages

The Standards Facility provides the flexibility to include optional, extended error messages in the Structured JCL Listing in addition to single-line, short error messages. The short message is specified in the EPLC-ERROR-MESSAGE field of the standards program. The extended text is stored in SJJSPARM library member DSSMSG90. This extended message can give users a better understanding of the violation beyond the basic short message. JOB/SCAN inserts the extended text into the SJL regardless of the JOB/SCAN mode used: ISPF panels, Edit macro, or batch.

### Preparing Extended Messages

#### *To prepare extended messages*

1. Place this line in the COBOL program after the MOVE-TO statements of the in-place error message.

```
MOVE 'X' TO EPLC-ERRTEXT-LVL
```

#### Preparing COBOL standards to insert extended text

```
IF EPLJ-CLASS NOT = 'P'
MOVE '608 CLASS PARAMETER NOT RIGHT'
  TO EPLC-ERROR-MESSAGE
MOVE 'X' TO EPLC-ERRTEXT-LVL

MOVE 01 TO EPLC-SEVERITY
PERFORM Z999-SET-ERROR THRU Z999-SET-ERROR-EXIT.
```

This turns on the extended message.

2. Compile the COBOL program.
3. Insert the message into the DSSMSG90 member of the SJJSPARM library using the same number from the message used in the COBOL program, prefixed with DSS9. Follow the instructions in the file.

#### Adding extended text to the User Messages member

```
003700 DSS9608 CLASS PARAMETER NOT RIGHT

003800 This JOB class is not right. Set it to "P" and revalidate.
```



- In the DSSMSG90 file, the DSS9xxx message number is required, but the short message text is not required. However, if the short message text is omitted in the COBOL standards, JOB/SCAN does not include it from the extended message file.
- The message number, which begins with 'DSS', must start in column 1, and 'DSS' must be in capital letters.
- The short message, when entered, should begin in column 11 on the same line as the error message number. The short message can be up to 48 characters in length.
- The extended text uses columns 2 through 72, can use an unlimited number of lines, and is completely free-form.
- Comments are indicated by an asterisk (\*) in column 1. They may appear anywhere within a message and JOB/SCAN does not include them in the SJL.
- Though ASG recommends that error message numbers be kept in sequence, it is not a requirement.
- ASG recommends that the severity letter (A, W, or E) be omitted from the error message number. It is not used by the JOB/SCAN Extended Message handler.

## Using Extended Messages

### To use extended messages

1. Follow instructions in [Default Error Message Handling](#) and [Preparing Extended Messages](#).
2. Specify the standards name and parameter according [Using the Standards Facility](#).
3. Run JOB/SCAN.
4. Examine the SJL. The sample in the below figure is a portion of the SJL from a foreground run showing the extended text from figure: [Preparing COBOL standards to insert extended text](#).

### SJL showing the extended text

```

STRUCTURED JCL LISTING      ASG, INC
DSNAME - CS082B.JCL      MEMBER - JSMVS
-----
|JOB - CS082BX                      MSGCLASS - X
|                                  CLASS - F
|PROGRAMMER NAME -NAME             ACCTG - ACCT
-----
| JOB JCL      | PROCLIB | SORT/UTILITY CARDS
-----
//CS082BX  JOB ACCT, 'NAME', NOTIFY=CS082B,
//          MSGLEVEL=1,MSGCLASS=X, CLASS=F,
//          REGION=0M, TIME=1440
**WARNING - DSS9608W - CLASS PARAMETER NOT RIGHT
- - - - -
This JOB class is not right. Set it to "P" and revalidate.
- - - - -
...

```

## Individual Control Over Standards Messages

If the Standards are to have individual control over multiple messages issued for the same statement type, then set the EPLC-REQUEST option to request control in one of these ways:

- In a COBOL Standards program:

```
MOVE 'Y' TO EPLC-REQUEST-INDIV-ERMSG-CNTL
```

- In an Assembler Standards program:

```
MVI C1RQINDM,C'Y' SET INDIVID CONTROL OVER MESSAGES
```

The Default for EPLC-REQUEST option is (N). With the default in force, suppose, for example, JOB/SCAN assigns multiple DSS9150 messages to one DD statement. If you mark this message for suppression, JOB/SCAN suppresses all the DSS9150 messages, and any change in severity to one affects the rest for the DD statement. However, with individual control asserted, JOB/SCAN suppresses or changes the severity of the targeted message occurrence only.

## Changing or Suppressing JOB/SCAN Error Messages

The EPL-JOB-SCAN-ERRORS table in the standards program contains the message number and severity level for every error message that JOB/SCAN issues for a statement. The standards program can change the severity of any JOB/SCAN message (except DSS0xxx error messages), or suppress the message altogether. (See the sample standards program CSMP in the SJSSTDL library for an example showing this function.)

Standards messages cannot be suppressed nor changed in severity; only JOB/SCAN system messages.



If the Suppress Error Messages Run Option is in effect, the system suppresses both JOB/SCAN and Standards Facility messages.

# Using Extended Parameter List Fields and Tables

The EPL allows standards to access the JCL parameters. Standards can test any JCL parameter by referring to it by an EPL name. The names all begin with EPL, followed by a one-character suffix designating the type of JCL statement (for example, J for JOB statement), a hyphen (-), and the name of the field. For example, the JOB name is referred to by the name EPLJ-JOBNAME; the DD name is referred to by the name EPLD-DDNAME.

- [EPL Example](#)
- [EPL Example Fields](#)
- [EPL Documentation](#)
- [Table Processing](#)
- [Using Control Table Entries](#)

## EPL Example

When a requirement is violated, then issue a message. A test typically consists of an IF statement that compares the JCL to the site requirement.

The requirement: The MSGCLASS parameter must be 'A'.

- The objective: If the MSGCLASS is not equal to 'A' then generate an error message.
- The code for the test: IF EPLJ-MSGCLASS NOT = 'A'



JOB/SCAN uses the name EPLJ-MSGCLASS for the MSGCLASS parameter of the JOB statement.

## EPL Example Fields

This table lists some commonly used EPL field names for JOB, EXEC, OUTPUT, PROC and DD statements. For a complete list of EPL fields for these and other types of JCL statements, browse or print the member CEPL from the JOB/SCAN SJSSTDL library.

JOB Fields	DD Fields	EXEC Fields	OUTPUT Fields	PROC Fields
EPLJ-JOBNAME	EPLD-DDNAME	EPLX-STEPNAME	EPLO OUTPUT NAME	EPLP-STEPNAME
EPLJ-CLASS	EPLD-UNIT	EPLX-PGM	EPLO-CLASS	EPLP-NAME-ON-EXEC
EPLJ-MSGCLASS	EPLD-DSNAME		EPLO-DEST	
EPLJ-NOTIFY	EPLD-RETPD			



JOB Fields	DD Fields	EXEC Fields	OUTPUT Fields	PROC Fields
EPLJ-REGION	EPLD-SPACE-PRIMARY EPLD-SPACE-SECONDARY			

## EPL Documentation

For more information about the format and contents of each EPL field, browse member EPLDOC in the SJSSTDL library. The below figure shows a portion of the EPLDOC member.

### Sample EPL Documentation in EPLDOC

```

EPLJ-CLASS                                PIC X.
Contains the job class coded with the CLASS parameter. A space in
this field indicates that the JOB statement did not contain a CLASS
parameter.

Example:      Code IF EPLJ-CLASS = SPACE      to determine if a CLASS
              parameter was coded.

Note: Even though a CLASS parameter coded on a JES3 /*MAIN statement
      overrides this CLASS parameter at run time, this field reflects
      the JOB statement value.

EPLJ-COMMENT                                PIC X(32).
Contains the first 32 characters of the comment field including
embedded blanks. This field may contain a JOB statement comment
or a short description from the SDF VSAM file. The Short Description
Run Option determines which takes precedence.

Example:      Code IF EPLJ-COMMENT = SPACES   to determine if any
              comments were provided for the JOB statement.

...

```

## Table Processing

Table handling within the Standards Facility allows the user to:

- Access user-defined JSOPT Control Tables.
- Analyze, suppress, or change the severity of JOB/SCAN errors.
- Evaluate control card input.

### What Is Involved in Using Tables?

The Standards Facility provides table-handling routines to access tables defined in the Extended Parameter List. Functions that can be performed on these tables include searching, sorting, reading, etc. These functions can be requested by passing a request code to a module named J06YTBL, which is provided on the product ship tape. The result is then passed back to the standards program as a return code. (See member CSMP in the SJSSTDL library for examples.) Return code values are described in this table:

Value	Description
0	Successful completion of table request.
4	An end of file occurred when sequentially reading an entry, or a duplicate key occurred when adding an entry.
8	An invalid request has been made.
12	A major problem with the table handling procedures has occurred.

### What Types of Tables Are Available?

This section describes the tables in the Extended Parameter. (For details, see the appropriate section of the EPLDOC member in the SJJSTDL library.)

#### User-Defined Control Tables (Ux)

User-defined Control Table entries are passed in the EPL-USER-RCT table. These tables hold lists of site-specific values that can be validated in the standards program. Because the actual table information is stored in the Control Table section of a PARMLIB member, the values are completely controlled by the user and are easy to modify. An example is included in the CALL-JOB-SECTION of member CSMP in the SJJSTDL library.

Control Tables reside in SJJSPARM library members JSOPT02 or JSOPT03, or in SYSIN DD statements containing lists of valid system codes, accounting parameters, DSN prefixes, etc. An advantage of using Control Tables to enter valid site conventions is that the validation list can be changed without modifying and recompiling the program. Another advantage is that different users can use different tables.

The Standards Facility provides table handling procedures that process user-defined Control Tables and control card input. These procedures are part of the table handling section of the Extended Parameter List that is compiled with the standards program.



REXX standards have a specialized function called JSTABLE. See [JSTABLE](#).

### JOB/SCAN Error Messages

Error messages issued by JOB/SCAN are passed for each statement in the EPL-JOB-SCAN-ERRORS table. Access to this table allows the standards program to suppress or change the severity of JOB/SCAN messages on a message-by-message basis. An example is included in the CALL-DD-ENTRY section of member CSMP in the SJJSTDL library.

#### Control Cards

JOB/SCAN processes control cards in two ways. The EPL processing option EPLC-CC-2003 determines how control cards and their associated DD statements are processed. The field is initially set to 'N' but can now be set to 'Y' by the Standards Program at EPLC-CALL-INITIAL time.

```
EPLC-CC-2003= 'N'
```

This is the default setting for this field. The DD statement is read and printed on the Structured JCL Listing. It is then parsed, but the EPLC-CALL-DD-ENTRY call to the Standards Program is not made until all control cards have been read for that DD. As each control card is read, it is passed to the Standards Program in EPLC-CONTROL-CARD at EPLC-CALL-CONTROL-CARD time. When end-of-file is reached, the EPLC-CALL-DD-ENTRY call is then made and the EPL-CONTROL-STATEMENTS table contains all of the control cards read for that DD statement. Any error messages associated with the DD statement have already been printed on the SJL and the severity cannot be modified nor the message suppressed. The AW Control Table will have to be used in this case.

```
EPLC-CC-2003= 'Y'
```

The other way, is when EPLC-CC-2003 is set to 'Y' the DD statement is read and printed on the Structured JCL Listing. It is parsed, and any error messages are tabled and then the EPLC-CALL-DD-ENTRY call to the Standards Program is made. The error messages are available to the Standards Program where the severity can be modified or the messages suppressed. No control card information is available at this time. Upon return from the Standards Program, the control cards are read and each control card is passed to the Standards Program in EPLC-CONTROL-CARD at EPLC-CALL-CONTROL-CARD time. When end-of-file is reached, the EPLC-CALL-END-CONTROL-CARDS call is made to the Standards Program and the EPL-CONTROL-STATEMENTS table contains all of the control cards associated with the previous DD statement.

## Using Control Table Entries

This topic describes about the format and example of a Control Table Entry.

### Format of a Control Table Entry

The format used by Control Table entries is:

Position	Value
1-5	Blank
6	U
7	Any character
8-72	A user-defined field. Only one value per entry is allowed. This field can begin in any position after position 7 and can contain up to 65 characters. It is recommended that the same format be used for each entry type.

### A Control Table Example

This is an example of Control Table UA. It defines valid site values in the JCL. This table can be found in JSOPT02.

```
.....+.....1.....+.....2.....+.....3.....
      UAPROD
      UATEST
      UAUPDT
      UACICS
```

UAIMS3

---

An example of how to access these Control Table values can be found in the CALL-JOB section of the CSMP sample standards program found in the SJJSTDL library.

# COBOL Standards Implementation Using the Prompter

This topic describes the use of the Standards Prompter panel (the Prompter) to build standards.

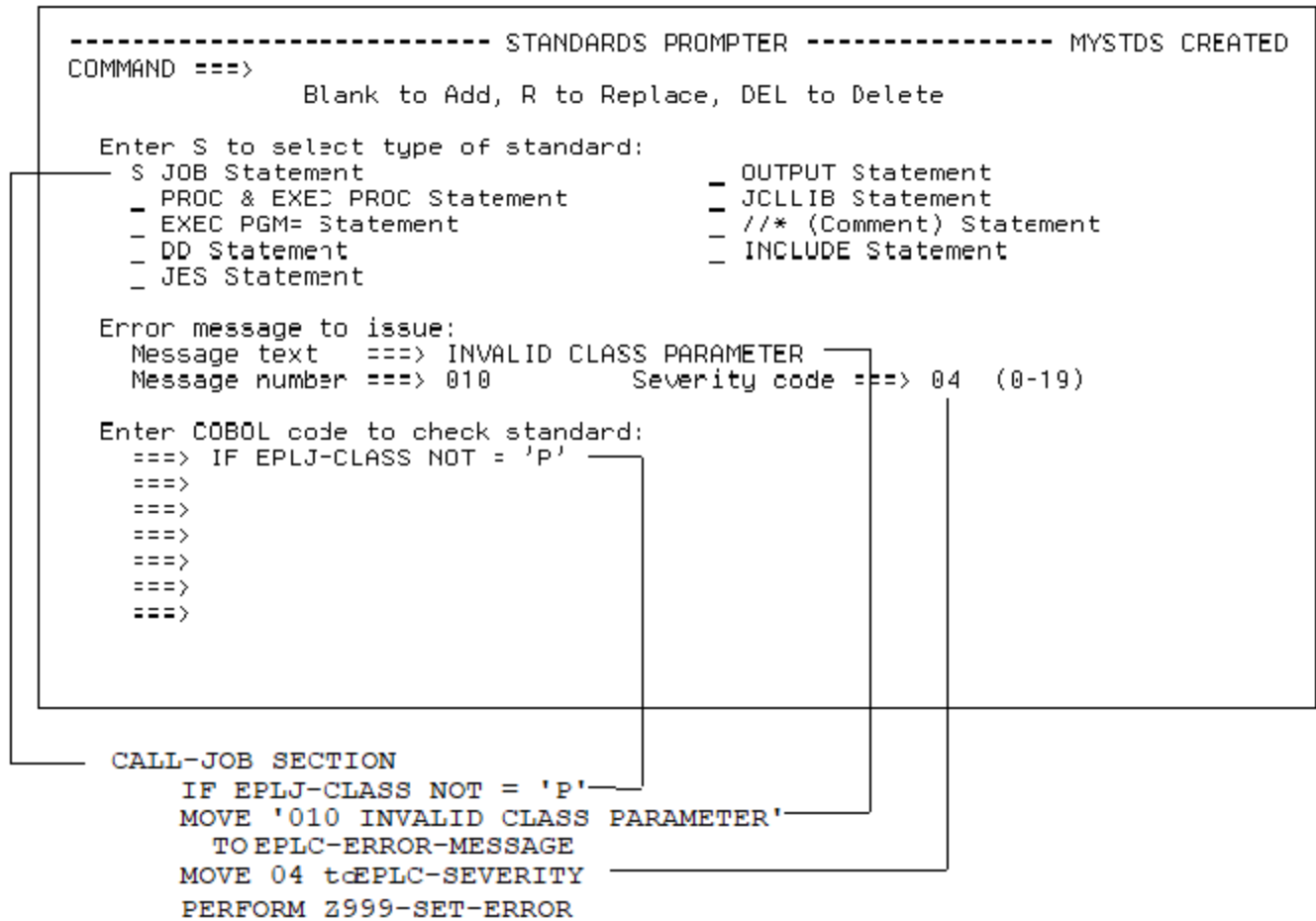
- [Prompter-Generated Standards](#)
- [Getting Started with the Prompter](#)
- [Creating and Maintaining Standards with the Prompter](#)
- [Preparing Standards Program for Execution](#)

## Prompter-Generated Standards

The Prompter facilitates the generation of COBOL standards, especially for first-time users or personnel unfamiliar with COBOL. As each requirement is coded on the Prompter, the Prompter inserts it into a copy of the SJSSTDL library member CTRKIT (named in the Member field of the Standards Utility Menu) in the appropriate SECTION. For example, JOB statement requirements are placed in the CALL-JOB SECTION and DD statement requirements are placed in the CALL-DD SECTION.

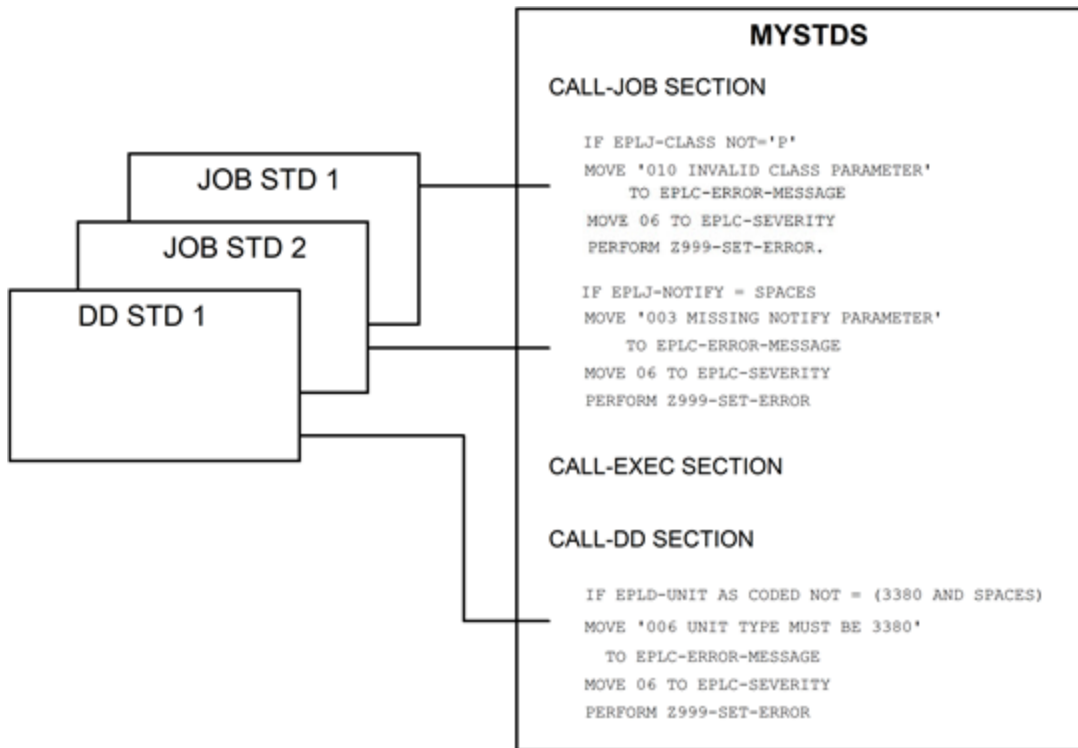
Figure: [MYSTDS Standards Placement](#) shows how the Prompter translates the fields into COBOL code. The contents of the field EPLJ-CLASS are compared to the letter 'P'. If the field and the letter 'P' don't match, the assigned message number '010' and the message text are moved to the special area designed to hold the error message (EPLC-ERROR-MESSAGE). The severity code '04', which means "Warning" to JOB/SCAN, is moved to the special area set aside for severity codes (EPLC-SEVERITY). PERFORM Z999-SET-ERROR passes the error message back to JOB/SCAN.

## MYSTDS Standards Placement



The below figure shows how a standards member is built by the Prompter.

## Anatomy of a Standards member



## Getting Started with the Prompter

ASG recommends using the Standards Utility Menu regardless of the specific approach chosen to write COBOL standards: whether the Prompter or through an editor. The Standards Utility Menu accommodates first-time users by providing access to the Standards Facility for creating, editing, compiling. An optional feature gives access to the compiler output. Ask your JOB/SCAN administrator about this option.

Before starting to write standards, be sure these conditions are met:

- The JOB/SCAN Standards environment is installed and you have access to it; check with your Product Administrator.
- You have access to either the JOB/SCAN SJSSTDL library or have the necessary members copied into your own library: CTRKIT, CEPL, CCHK, and CERR.



Never modify the SJSSTDL library master copies of CTRKIT, CEPL, CCHK, or CERR directly. Work with the Prompter-generated copy or a copy in another library.

## Invoking the Standards Utility Menu

### To invoke the Standards Utility Menu

1. Initialize JOB/SCAN.
2. Select the Standards Utility Menu option from ASG-JOB/SCAN Primary Menu.

If the Standards Utility Menu does not appear as one of the options on the ASG-JOB/SCAN Primary Menu, ask your Product Administrator.

### Invoking the Standards Utility Menu

```

----- ASG-JOB/SCAN PRIMARY MENU -----
SELECTION ==> 4

  0 - Hardcopy Output and JOB Statement Definition
  1 - Validate JCL
  2 - Change / Reformat JCL
  3 - Processing Options
  4 - Standards Utility Menu
  5 - Product Administration Menu
  X - Exit

USERID - USER01
PREFIX - USER01
DATE - 2009/02/23
TIME - 12:54
APPLID - JS

```

The Standards Utility Menu is the starting point for creating and maintaining standards; from it you create, edit and compile standards. The below figure shows the initial view of the Standards Utility Menu. Initialize the Standards Source Library and member fields. This member is where the Prompter stores standards statements; then initialize the Prompter libraries.

### Standards Utility Menu

```

----- STANDARDS UTILITY MENU -----ASG-JOB/SCAN
OPTION ==>

  S  Setup      - Specify Prompter Parameters
  P  Prompter   - Use Prompter to Create/Modify Standards Program
  E  Edit       - Edit Standards Program using ISPF Edit
  C  Compile    - Compile Standards Program
  X  Exit       - Return to ASG-JOB/SCAN Menu
  X;1 ASG-JOB/SCAN - Display the ASG-JOB/SCAN JCL Validation Menu

STANDARDS SOURCE LIBRARY ==>
(Name of PDS to contain the COBOL source code of the Standards Program)

      MEMBER ==>          (Standards Program name or
                          blank for member selection list)

```



## Initialize the Standards Source Library and Member Fields

### To set the Standards Source Library field

- At the bottom of the Standards Utility Menu, fill in the Standards Source Library field with the name of the dataset where the Prompter puts the standards COBOL program it generates. It can be any PDS with 80 byte records.

### To set the Member field

- Choose one:
  - Fill in the Member field with the name of existing or new standards.
- Or**
- Leave the Member field blank and type P (for Prompter), E (for Edit), or C (for Compile) in the command line and press Enter to select a member from the list. For Edit and Compile, the prompter executes that function prior to returning to the Standards Utility Menu.

When both fields are complete, press PF3 to return to the Standards Utility Menu. JOB/SCAN saves the values in your profile for subsequent use.

The below figure shows the Standards Utility Menu with the Standards Source Library and Member fields completed. As you become more comfortable with writing standards, you might find that editing the Standards Program directly with ISPF Edit provides additional flexibility. The 'E' option on the Standard Utility Menu provides convenient access to ISPF Edit.



Using the Prompter to modify any COBOL program it did not generate can yield unpredictable results. It is best to use Prompter only with code it generates.

### Example Library and Member Fields

```
----- STANDARDS UTILITY MENU -----ASG-JOB/SCAN
OPTION  ===>

      S  Setup      - Specify Prompter Parameters
      P  Prompter   - Use Prompter to Create/Modify Standards Program
      E  Edit       - Edit Standards Program using ISPF Edit
      C  Compile    - Compile Standards Program
      X  Exit       - Return to ASG-JOB/SCAN Menu
      X;1 ASG-JOB/SCAN - Display the ASG-JOB/SCAN JCL Validation Menu

STANDARDS SOURCE LIBRARY ===> COBOL
(Name of PDS to contain the COBOL source code of the Standards Program)

      MEMBER ===> PROGVAR (Standards Program name or
                          blank for member selection list)
```

## Initialize the Prompter Libraries

Before you use the Standards Utility Panel for the first time, a number of dataset names and a JOB statement must be specified. The Standards Prompter can then generate standards programs. Enter these parameters on the Specify Prompter Parameters panel. See figure: [Panel for Setup option on Standards Utility Menu](#).

Use the CANCEL (CAN) on the command line to exit without saving the values.

### To specify Prompter parameters

1. Select option *S* Setup from the Standards Utility Menu.  
Initially, the Options Panel is displayed without any fields completed.
2. Press Enter to cause the Prompter to fill in as many fields as it can. Usually, it fills in the STDLIB, JOB/SCAN Load Library, Standards Load Library, and Temporary Work Unit fields.
3. Check that the values JOB/SCAN places in these fields are the ones you want and change as necessary.

#### ASG-JOB/SCANSJJSSTD Library

The JOB/SCAN SJJSTDL library is one of the libraries downloaded and installed with the product files. It contains several members used by the Prompter including CTRKIT, CEPL, CERR, and CCHK. See [SJJSTDL Library Members](#) for more about these members.

Enter the name of the JOB/SCAN SJJSTDL library installed at your site. For example:

```
JOB/SCAN STDLIB ====> 'ASG.$PRD.SJJSTDL'
```

#### ASG-JOB/SCAN Load Library

Enter the name of the JOB/SCAN load library (SJJSLMOD) in the JOB/SCAN Load Library field. The JOB/SCAN load library is included in the product installation files. For example:

```
ASG-JOB/SCAN LOAD LIBRARY ====> 'ASG.@UTL.SJJSLMOD'
```

#### STANDARDS LOAD LIBRARY

Enter the name of the load library that you want the standards program to be compiled into and executed from. It can be the JOB/SCAN load library or any valid load library. But, if not the JOB/SCAN load library, it might have to be allocated under the DD name JLLD for the standards program to execute when JOB/SCAN runs. See [Reference Information](#) for additional information on JLLD. For example:

```
STANDARDS LOAD LIBRARY ====> 'ASG.$DEV.SJJSLMOD'
```

#### TEMPORARY WORK UNIT

TEMPORARY WORK UNIT is the DASD unit name for temporary work datasets. The default is SYSDA. If your site's unit name for allocating temporary datasets is different, change this field to the valid unit name.

```
TEMPORARY WORK UNIT ====> SYSDA
```

4. Fill in the remaining fields.

#### JOB Statement

The first five input lines on the Specify Prompter Parameter panel are for the JOB statement. Enter a valid JOB statement. Code unneeded lines as comments, beginning with `//*`.

For example:

```
====> //ASGCOM2 JOB (ASGICS,132900,ASG,37), 'USER',
====> //          NOTIFY=ASGCOM2
====> /*XEQ NJEZ
====> /*ROUTE PRINT ASGI101
====> /*
```

### COMPILE STEPLIB

The COMPILE STEPLIB is the name of the load library containing the COBOL compiler. The name of this library can be obtained from the STEPLIB DD statement in a COBOL compile PROC or JCL. For example:

```
COMPILE STEPLIB ====> 'SYS1.VSCOBOL.COMPILER'
```

The library name is not required if the COBOL compiler is in the LNKLSTxx. Leave the field blank and the program does not generate a STEPLIB DD.

### LINK EDIT SYSLIB

The LINK EDIT SYSLIB is the name of the load library containing the COBOL subroutines. The name of this library can be obtained from the SYSLIB DD statement in a link edit PROC or JCL. For example:

```
LINK EDIT SYSLIB ====> 'SYS1.VSCOBOL.COBLIB'
```

5. After filling in the remaining fields, press Enter to validate them, or press the END PF key to validate and save the values, then exit. JOB/SCAN does not save invalid values.

### Panel for Setup option on Standards Utility Menu

```
----- SPECIFY PROMPTER PARAMETERS -----ASG-JOB/SCAN
OPTION  ====>

Job statements for batch compile.
====> //JOBNAME JOB (ACCOUNT), 'USER NAME          ',
====> //          NOTIFY=USER01
====> /*
====> /*
====> /*

ASG-JOB/SCAN STDLIB          ====> 'ASG.SJJSSTD '
ASG-JOB/SCAN LOAD LIBRARY   ====> 'ASG.@UTL.SJJS LMOD'
STANDARDS LOAD LIBRARY      ====> 'ASG.$STG.SJJS LMOD'
COMPILE STEPLIB             ====> 'ASG.STEPLIB
LINK EDIT SYSLIB            ====> 'ASGLIB.SYSLIB'
TEMPORARY WORK UNIT         ====> SYSDA
```

6. Correct any errors. Save and exit.

## Creating and Maintaining Standards with the Prompter

You have two choices to write standards:

- Use the Prompter.
- Use the Edit option from the Standards Utility Menu to directly work with COBOL in ISPF Edit. If you do this, specify the Standards Source Library name and the Member on the Standards Utility Menu. If it is a new standards program, you may copy CTRKIT as a starter kit.



Do not directly modify the original copy of CTRKIT; make a copy first.

This section describes creating and maintaining COBOL standards using the Prompter. Before you can use the Prompter, make sure you have completed the setup steps in [Getting Started with the Prompter](#).

The Prompter panel (called the Prompter) builds standards only in COBOL from values you enter into its fields. Generally, each JCL requirement beyond basic syntax and catalog checking, for instance particular message classes or programmer names, is individually input into the Prompter. The collection of these codified individual requirements is called standards, and each individual one as it is codified into COBOL is called a standard.

### Blank Standards Prompter

```

----- STANDARDS PROMPTER ----- MYSTDLIB LOADED
COMMAND ===>
      Blank to Add, R to Replace, DEL to Delete

Enter S to select type of standard:
  _ JOB Statement                _ OUTPUT Statement
  _ PROC & EXEC PROC Statement   _ JCLLIB Statement
  _ EXEC PGM= Statement          _ /* (Comment) Statement
  _ DD Statement                 _ INCLUDE Statement
  _ JES Statement

Error message to issue:
Message text   ===>
Message number ===>          Severity code ===>      (0-19)

Enter COBOL code to check standard:
===>
===>
===>
===>
===>
===>
===>

```

### Adding a New Standard

#### To add a new standard

1. Type P on the Standards Utility Menu and input the source library and member and press Enter to access the Prompter.
2. Type S by the type of standard you want to add (JOB statement, for example).



Use Edit to write a standard for a statement type not listed in the Prompter.

3. Fill in the Message text, Message number, Severity code, and COBOL code fields.
4. Press Enter. JOB/SCAN validates the values in the fields and adds the new standard to the COBOL program.



JOB/SCAN does not add a new standard if the message number is a duplicate.

### Standards Prompter

```

----- STANDARDS PROMPTER -----
COMMAND ==>>
          Blank to Add, R to Replace, DEL to Delete

Enter S to select type of standard:
  _ JOB Statement                _ OUTPUT Statement
  _ PROC & EXEC PROC Statement  _ JCLLIB Statement
  _ EXEC PGM= Statement         _ /* (Comment) Statement
  _ DD Statement                _ INCLUDE Statement
  _ JES Statement

Error message to issue:
Message text ==>> INVALID CLASS PARAMETER
Message number ==>> 018          Severity code ==>> 04 (0-19)

Enter COBOL code to check standard:
==>> IF EPLX-PGM = 'IDCAMS'
==>>
==>>
==>>

```

When the Prompter adds a standard, it refreshes the panel by:

- Clearing the Message text and COBOL code fields.
- Retaining the Message number for ready reference.
- Retaining the Severity code so that it doesn't have to be re-entered.
- Retaining the type of selected standard with S in the selection field.



Consider these points while using the Prompter:

- Only one standards type can be selected at a time.
- The Message field is 48 characters long.
- The Message Number field is validated for number and duplicates.
- The Severity Code is validated for number and range from 0 to 19.
- JOB/SCAN validates any fields prefixed with 'EPL' in the COBOL code fields against the EPL field names list.
- The Prompter has room for 7 lines of COBOL; if you need more, you may edit the code and make changes, but when returning to the Prompter, it only displays the first seven lines for a standard. Remember, using the Prompter to modify COBOL it did not generate can lead to unexpected results.

The below figure shows how to add more than one line of text.

### Three lines of text in the COBOL code field

```

----- STANDARDS PROMPTER -----
COMMAND ==>
      Blank to Add, R to Replace, DEL to Delete

Enter S to select type of standard:
  _ JOB Statement                _ OUTPUT Statement
  _ PROC & EXEC PROC Statement  _ JCLLIB Statement
  _ EXEC PGM= Statement         _ /* (Comment) Statement
  S DD Statement                _ INCLUDE Statement
  _ JES Statement

Error message to issue:
Message text ==> YOU MAY NOT USE STEPLIB FOR IDCAMS
Message number ==> 120          Severity code ==> 04 (0-19)

Enter COBOL code to check standard:
==> IF EPLX-PGM = 'IDCAMS'
==> AND
==> EPLD-DDNAME = 'STEPLIB'
==>
==>
==>
==>

```

### Prompter Errors

If the Prompter detects an error in the field values, it displays an error message in its top, right corner. If this message does not clarify the error, press PF1 for help. Pressing PF1 once displays a longer message and pressing PF1 again displays the full help screen. For example, if you entered IF EPLJ-CLAS NOT = 'P', the Prompter displays the message FIELD NAME NOT FOUND because you misspelled EPLJ-CLASS. Refer to the below figure for an example of extended help from the COBOL code field.

### Extended Help for COBOL Code Field

```

----- STANDARDS PROMPTER ----- ASG-JOB/SCAN
OPTION ==>

      |-----|
      | EPL FIELD NAMES |
      |-----|

Some commonly used EPL fields are listed below. For a complete list, see
the ASG-JOB/SCAN Reference Guide or EPLDOC in 'ASG.PROD.SJSSTD'

JOB Statement Fields      PROC Statement Fields      DD Statement Fields
EPLJ-JOBNAME              EPLP-LEVEL-NUMBER          EPLD-DDNAME
EPLJ-JOBNAME-C (n)       EPLP-STEPNAME              EPLD-BLOCKSIZE-AS-CODED
EPLJ-ACCOUNTING-DATA     EPLP-STEPNAME-C (n)       EPLD-LRECL-AS-CODED
EPLJ-CLASS                EPLP-NAME-ON-EXEC          EPLD-MODEL-DSCB
EPLJ-MSGCLASS             EPLP-ACCOUNTING            EPLD-RECFM-AS-CODED
EPLJ-NOTIFY               EPLP-ACCOUNTING            EPLD-DISP-1-AS-CODED
EPLJ-PROGRAMMER-NAME     OUTPUT Statement Fields    EPLD-DSNAME
EPLJ-REGION               EPLO-OUTPUT-NAME           EPLD-DSN-NODE (n)
                           EPLO-CLASS                  EPLD-RETPD

```

## Replacing Standards

### To replace standards

1. Select the standards type with S and press Enter. JOB/SCAN takes the first standard of the type from the COBOL program and places its values into the fields.
2. Scroll the standards, if necessary.
3. Make the changes to the fields on the Prompter panel.
4. Type R (for Replace) on the command line.
5. Press Enter.



If you change the fields on the panel and press Enter without putting an R on the command line, the Prompter attempts to add a new standard.

## Deleting Standards

### To delete standards

1. Select the standards type with S and press Enter. JOB/SCAN takes the first standard of the type from the COBOL program and places the values in the fields.
2. Scroll the standards, if necessary.
3. Type DEL on the command line.
4. Press Enter.

## Other Prompter Functions

These additional functions are available from the Prompter.

### Finding a Standard

The FIND command searches the standards for specified strings. For example, find the next occurrence of EPL field name EPLD-LRECL by typing

```
FIND EPLD-LRECL
```

Find the first occurrence of MOVE EPLJ-JOBNAME, by typing:

```
FIND FIRST 'MOVE EPLJ-JOBNAME'
```

Put quotes around strings that contain blanks. Valid parameters for the FIND command are FIRST, LAST, NEXT, and PREV with NEXT being the default.



The FIND command searches from the last found target. So if you search for a string that is in a line above the last found string, FIND will not find the string. For example in this code:

```
EPLJ-CLASS NOT = 'P'
MOVE '808 THIS IS ONLY A TEST'
    TO EPLC-ERROR-MESSAGE
MOVE 04 TO EPLC-SEVERITY
PERFORM Z999-SET-ERROR THRU Z999-SET-ERROR-EXIT.
```



```
IF EPLJ-CLASS NOT = 'Z'
MOVE '806 THIS IS ANOTHER TEST AGAIN'
  TO EPLC-ERROR-MESSAGE
MOVE 04 TO EPLC-SEVERITY
PERFORM Z999-SET-ERROR THRU Z999-SET-ERROR-EXIT.
```

searching for 808 after finding 806 results in the STRING NOT FOUND message.

## Scrolling through Standards

The Prompter stores the code for each type of JCL statement in the appropriate area of the standards. For example, all statements relating to the JOB statement are stored in the CALL-JOB SECTION. Use the UP and DOWN commands on the command line to scroll through the standards in a section (or press the corresponding PF keys (usually PF7 and PF8)).

Scroll the standards of a different section by selecting the JCL statement type with S and using the UP and DOWN commands.

## Saving Standards

Enter `SAVE` on the command line of the Standards Prompter panel. Or press PF3 to save and exit.

## Exiting the Prompter (and Saving Values)

Press PF3 or type `END` on the command line. If the standards were not already saved, the Prompter saves them before exiting.

## Exiting without Saving

Enter `CANCEL`. If standards were not saved, the Prompter exits without saving the standards.

## Getting Help

Press PF1 to display the short help and press PF1 again to display the extended help anytime you need help with the Prompter.

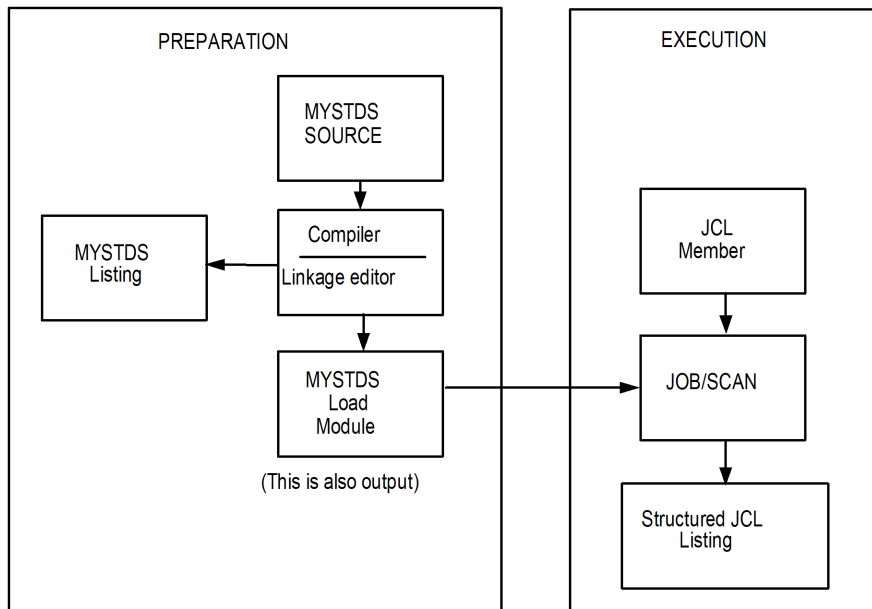
## Executing Standards from Prompter

Run JOB/SCAN with standards using the member on the Standards Utility Menu, using the `X;1` option from the Standards Utility Menu. This command displays the ASG-JOB/SCAN JCL Validation panel and writes the name of your Standards to the execution panel. Select a JCL member and execute JOB/SCAN in the foreground.

## Preparing Standards Program for Execution

Standards built by the Prompter become input to `COMPILE/LINKEDIT`. The `COMPILE/LINKEDIT` job produces a set of listings and, if there are no errors, produces a ready-to-run load module.





### Compiling Standards

The Compile option on the Standards Utility Menu submits a batch job to the system. When the job is completed, the return code from the compile and link steps are displayed on your screen. If both steps return with a code of 00, proceed to test your standards. If the compile step results in a higher return code, find and correct the errors. For additional information, see [COBOL Standards Testing and Debugging](#).

### Reviewing the Compiler Output

If the compile step did not execute with a return code of 00, browse the output to see what is wrong. Use the 'O' option, if installed, on the Standards Utility Menu to invoke SDSF, or other previously set up utilities.

# COBOL Primer

This topic introduces the COBOL language, which is used to write a standards program.

- Introduction to COBOL
- Words and Symbols
- Statements
- Sentences
- Comments
- Paragraphs and Sections
- Standards Program Sections
- Program Divisions
- Format of a Standard
- Condition
- Message and Message Number
- Severity Code
- Z999-SET-ERROR
- Indentation
- Simple Comparisons (=, >, <, NOT)
- Testing Single Character Fields
- Testing CLASS
- The Difference Between 'AS CODED' and 'AS USED' EPL Fields
- Compound Comparisons
- Using AND and OR
- Testing Two Different JCL Fields
- Testing One Character in a Multi-Character Field
- Testing a Group of Characters in a Multi-Character Field
- Testing a Group of Characters in a Field
- Using the SUBSTRING Functions
- Testing Numeric and Character Fields
- Testing JCL Parameters
- Adding Fields to WORKING-STORAGE
- Field Names
- Level Numbers
- Pictures

- Testing Dataset Name Nodes
- Using UNSTRING
- Counting Items
- Standards Requiring Calculations
- Using the CALL-\*SECTIONS
- Multiple IFs and ELSEs
- Defining External Tables
- Using Dynamic Tables
- Advanced Topics
- Arithmetic
- Generating JCL Change Commands with the Standards Program

## Introduction to COBOL

COBOL is an English-like computer language. Even if you have never seen COBOL before, you can probably understand some parts of a COBOL program. For example, you might be able to guess that the COBOL statement:

```
IF EPLJ-JOBNAME = 'TESTJOB'
```

tests whether the JOB name is 'TESTJOB'. COBOL is a powerful language with many features, though many of these features are not needed to write standards. The subset of COBOL this topic describes is enough to write most standards. If you want to learn more, there are many guides to COBOL and several IBM manuals that explain in detail the features of the language.

## Words and Symbols

At the lowest level, COBOL is made up of words and symbols separated by spaces. In the previous example, 'IF' and 'EPLJ-JOBNAME' are words; '=' is a symbol. Some words are predefined by COBOL ('IF' in the previous example). These are called reserved words.

Some words are predefined by JOB/SCAN ('EPLJ-JOBNAME' in the previous example). You might want to define other words in your standards. (User-defined words are typically used as the names of fields in which to save information or perform calculations.) These are the valid characters for user-defined words:

- A through Z
- 0 through 9
- - (hyphen)

COBOL also predefines some symbols. This table lists predefined COBOL symbols.

Symbol	Description
'	quote mark
.	period
=	equals
<	less than
>	greater than
(	left parenthesis
)	right parenthesis

## Statements

A statement is any valid combination of COBOL words and symbols beginning with a COBOL verb such as 'IF' or 'MOVE'. For example:

```
IF EPLJ-JOBNAME = 'TESTJOB'
```

is a statement. Statements must begin in column 12 or later on a line in a COBOL program. (In COBOL, the area from column 12 to column 72 is called 'Area B'.)

## Sentences

A COBOL sentence consists of one or more statements ending with a period. Adding a period at the end of the statement in the previous example would make it a sentence. However, as in English, a period in the wrong place can change the meaning of what you write. Consider the example:

```
IF EPLJ-JOBNAME = 'TESTJOB'.
    MOVE '100 INVALID JOB NAME' TO EPLC-ERROR-MESSAGE.
```

The period after the IF statement terminates it prematurely. Because the MOVE statement is in a different sentence than the IF statement, it does not depend on the job name tested by the IF statement. The MOVE statement always fills in the error message field, regardless of the job name. Because they consist of statements, sentences must begin in column 12 or later on a line in a COBOL program.

If a statement cannot fit on one line, you may break it up into multiple lines. For example:

```
MOVE '100 THIS JOB NAME IS INVALID'
    TO EPLC-ERROR-MESSAGE.
```

## Comments

ASG recommends adding comments to explain the standards. In COBOL, a line with an asterisk (\*) in column 7 is considered a comment line. Anything may be written on the line after the asterisk.

The COBOL compiler ignores columns 1-6. Those columns can be used for a brief comment on any line, whether or not there is an asterisk in column 7.



Use caution when modifying a Prompter-generated COBOL program. If you change the header or standards code, the Prompter will not be able to insert new standards.

## Paragraphs and Sections

In most programs, there are groups of sentences that are used repeatedly. A COBOL standards program, for example, might have sentences that are executed for every DD statement. These sentences are kept together as a group.

In COBOL, sentences might be grouped into paragraphs or sections. Optionally, paragraphs may be grouped into sections. Paragraph names begin with a user-defined word followed by a period. Section names begin with a user-defined word followed by the COBOL reserved word SECTION.

Example paragraph name: DD-NAME-TESTS.

Example section name: CALL-END-STEP SECTION.

JOB/SCAN Prompter-generated standards programs don't normally use paragraphs. Instead, sentences are grouped into JOB/SCAN-defined sections. Paragraph names and section names must begin in column 8, 9, 10, or 11. These columns are known as 'Area A' in COBOL.

## Standards Program Sections

There is a section in COBOL standards program for each type of JCL statement. The name of each of these sections begins with 'CALL-' followed by the name of the type of JCL statement. For example, the section that processes JOB statements is called:

```
CALL-JOB SECTION.
```

Place tests that apply only to certain JCL statements in the corresponding section. Thus all JOB statement tests should be in the CALL-JOB SECTION and all DD statement tests should be in the CALL-DD-ENTRY SECTION.

In addition to the JCL statement sections, there are sections that are executed at the beginning or end of a group of JCL statements. These include sections executed before the first statement, at the end of each step, at the end of each PROC, and at the end of the job. For example:

```
CALL-INITIAL SECTION.
CALL-END-STEP SECTION.
CALL-END-PROC SECTION.
CALL-END-JOB SECTION.
```

Code in these sections is used with code in other sections to check for the presence of something in a group of JCL statements. For example, to test whether there is a SYSUDUMP card in a step, code would be written in the CALL-PROGRAM, CALL-DD-ENTRY, and CALL-END-STEP SECTIONS.

## Program Divisions

Every COBOL program is divided into four divisions. This table list the divisions:

Division	Description
Identification Division	Name of the program, create, and update dates
Environment Division	File definitions
Data Division	Data definitions
Procedure Division	Executable COBOL statements

CTRKIT contains sample code for the four basic program divisions. If you are using the Prompter, you should never modify CTRKIT (because the Prompter uses CTRKIT). However, you can make a working copy of CTRKIT.

In the Identification Division, the program name should match the member name of the COBOL standards program. You might also wish to change some of the other lines in the Identification Division, such as the update date and the programmer name. The Environment Division should not need to be changed. If you want to define your own data areas to save information or perform calculations, put them in the Data Division. The Procedure Division includes the sections that process the various types of JCL statements. Put executable COBOL statements in the appropriate sections of the Procedure Division.

## Format of a Standard

On the Prompter ISPF panels, there are blanks to fill in to define each part of the standard (see [COBOL Standards Implementation Using the Prompter](#)). When not using the Prompter, you must include each of these parts in the COBOL statements you write. In addition, you must execute a special section in the Standards program to issue the message you wish to generate.

## Condition

The condition is generally in the form of an IF statement. For example:

```
IF EPLJ-JOBNAME-C (1) NOT = 'P'
```

## Message and Message Number

The message is built in a data area called 'EPLC-ERROR-MESSAGE'. Both the message number and the message text are moved into this area as a single text string in quotes. For example:

```
MOVE'101 NOT A VALID PRODUCTION JOB NAME'
  TO EPLC-ERROR-MESSAGE
```

If the message number is omitted from the message string, JOB/SCAN uses '999'.

## Severity Code

The severity code is placed in a field named 'EPLC-SEVERITY'. For example:

```
MOVE08 TO EPLC-SEVERITY
```

## Z999-SET-ERROR

The last statement of each standard should execute the Z999-SET-ERROR SECTION to issue the message to the user. To execute the section, use the COBOL verb PERFORM. For example:

```
PERFORM Z999-SET-ERROR.
```

## Indentation

Putting the parts of the standard together results in the COBOL sentence:

```
IF EPLJ-JOBNAME-C (1) NOT = 'P'
  MOVE '101 NOT A VALID PRODUCTION JOB NAME'
    TO EPLC-ERROR-MESSAGE
  MOVE 08 TO EPLC-SEVERITY
  PERFORM Z999-SET-ERROR.
```



The only period is at the end of the sentence. Also, each statement after the initial IF statement is indented to show that these other statements are part of the IF statement.

Similarly, 'TO EPLC-ERROR-MESSAGE' is indented to show that it is part of the previous MOVE statement. Indentation is not required by COBOL, but is often done to emphasize organization, making it easier for programmers to read and find errors in the code.

## Simple Comparisons (=, >, <, NOT)

Many standards compare a JCL parameter to a text string. This is written as an IF statement in COBOL. IF statements in COBOL can test whether one field is equal to, greater than, or less than another field by using the symbols: =, >, <. For example:

```
IF EPLD-DDNAME = 'SYSUDUMP'
```

IF statements can also test whether a field is not equal to, not greater than, or not less than another field by adding the word NOT before the symbol: NOT =, NOT>, and NOT<. For example:

```
IF EPLJ-MSGCLASS NOT = 'A'
```

## Testing Single Character Fields

JCL fields may contain a single character. To test these fields, write an IF statement that compares the JCL field (using the EPL field name) with a character in single quotes. See examples in [Testing CLASS](#).

### Testing CLASS

Assume that 'A' is not a valid class. The standard should issue a message if the class specified in the JCL is 'A'.

If you are using the Prompter, fill in the message and severity code input areas, and enter this COBOL statement in the COBOL code area of the ISPF panel:

```
IF EPLJ-CLASS = 'A'
```

### Testing MSGCLASS

Assume that 'P' is the only valid MSGCLASS. The standard should issue a message if the MSGCLASS specified in the JCL is not 'P'.

```
IF EPLJ-MSGCLASS NOT = 'P'
```

### Testing Multi-Character Fields

Testing multi-character fields is very similar to testing single character fields. The only difference is that the comparison string is longer.

The string is compared with the whole JCL field. For example, the comparison:

```
IF EPLJ-JOBNAME = 'TEST'
```

is true only when the job name is exactly equal to TEST. It is not true if the job name begins TEST, (for example, TEST01).

Testing parts of multi-character fields is covered in [Testing One Character in a Multi-Character Field](#)

### Testing a DD Name

Assume that 'SYSUT3' is not a valid DD name. The standard should issue a message if the DD name in the JCL is 'SYSUT3'.

```
IF EPLD-DDNAME = 'SYSUT3'
```

### Testing the Job Name

Assume that 'TESTJOB1' is not a valid job name. The standard should issue a message if the job name in the JCL is 'TESTJOB1'.

```
IF EPLJ-JOBNAME = 'TESTJOB1'
```

## The Difference Between 'AS CODED' and 'AS USED' EPL Fields

A parameter might have one value as coded in the JCL and a different value when the job is executed. This happens when the parameter is overridden by another parameter or when a symbolic is substituted with a value. For example, the REGION parameter specified on a JOB statement can override the REGION parameter specified on the EXEC statement.



For parameters of this type, the Standards program can test either the value coded in the JCL or the value actually used when the job executes. There are two fields in the EPL for these parameters. One of the field names has the suffix '-AS-CODED' to indicate that it contains the value coded in the JCL. The other one contains the value used during job execution.

### Testing the REGION Size

A REGION parameter on the JOB statement might override a REGION parameter on the EXEC statement. This statement tests if the REGION parameter is coded on the EXEC statement:

```
IF EPLX-REGION-AS-CODED NOT = SPACES
```

This statement tests if the REGION was either specified on the EXEC statement or was an override from the JOB statement:

```
IF EPLX-REGION NOT = SPACES
```

### Testing the COND Parameter

This statement tests if COND was not coded on the EXEC statement:

```
IF EPLX-COND-AS-CODED NOT = SPACES
```

This statement tests if there is no COND in effect for the step, regardless of whether it was specified on the EXEC statement or the JOB statement:

```
IF EPLX-COND NOT = SPACES
```

### Testing DISP on a DD Statement

This statement tests if the DISP parameter contains 'UNCATLG' in the second position, for example, DISP = (OLD, UNCATLG):

```
IF EPLD-DISP-2-AS-CODED = 'UNCATLG'
```

## Compound Comparisons

Previous examples have shown how to test a single JCL field for a single value. However, there are often several different valid values for a JCL field. For example, a site might allow production jobs to use MSGCLASS A, R or Z.z

There are two ways to write this kind of comparison in COBOL. One way is to write multiple comparisons combined with ANDs or ORs. The other way is to write a single comparison that compares a JCL field with a list of values.

### Using Logical Operator AND

These statements test if MSGCLASS is not an A, R, or a Z:

```
IF EPLJ-MSGCLASS NOT = 'A'  
AND EPLJ-MSGCLASS NOT = 'R'  
AND EPLJ-MSGCLASS NOT = 'Z'
```

A shorter way to write the same test:

```
IF EPLJ-MSGCLASS NOT = ('A' AND 'R' AND 'Z')
```

(Read 'If MSGCLASS is not A, and MSGCLASS is not R, and MSGCLASS is not Z'.)

## Using Logical Operator OR

These statements test if the job name is TESTJOB1 or TESTJOB2:

```
IF EPLJ-JOBNAME = 'TESTJOB1' OR
EPLJ-JOBNAME = 'TESTJOB2'
```

Another way to write the same test:

```
IF EPLJ-JOBNAME = ('TESTJOB1' OR 'TESTJOB2')
```

(Read 'If job name is TESTJOB1, or job name is TESTJOB2'.)

## Using AND and OR

If the comparison contains a NOT, use AND, otherwise, use OR. The two ways of writing compound comparisons (multiple comparisons and a single comparison with a list of values) give the same results. Use whichever approach is easiest for you.

If the list of values is long, or if the values change frequently, you might want to use a Control Table instead of a list of values. See [Defining External Tables](#) for more information.

## Testing Two Different JCL Fields

The previous examples showed how to compare a single field with several values using AND and OR. A standard might require tests on two different fields. This can be done by combining two comparisons with an AND or an OR.

### Testing that Certain Units are Not Specified on New Datasets

This standard issues a message if DISP=NEW is specified and the specified UNIT is not DISK, SYSDA, or SYSSQ. Add this code to the CALL-DD-ENTRY SECTION:

```
IF EPLD-DISP-1-AS-CODED = 'NEW' AND
EPLD-UNIT-AS-CODED = ('DISK' OR 'SYSDA' OR 'SYSSQ')
```

## Testing One Character in a Multi-Character Field

A standard might restrict a particular character in a JCL field (such as job name) to a set of valid values. Individual characters may be directly accessed in some EPL character fields, including job name, DD name, and dataset name. To access a particular character, the suffix -C is added to the field name and an index number is specified in parentheses. The index number tells which character to compare.

In the below figure, you can see how the JOB name from the JOB statement is placed in the EPLJ-JOBNAME field. Because there are 8 characters in this field, the valid index numbers are 1 through 8.

**Contents of EPLJ-JOBNAME**`//PJGL0020 JOB (...)`

EPLJ-JOBNAME	P	J	G	L	0	0	2	0
	1	2	3	4	5	6	7	8

EPLJ-JOBNAME-C (1) ———┐  
 EPLJ-JOBNAME-C (3) ———┘

**Testing a Single Character in the Job Name**

Assume that the first character in the job name must be 'D'. The standard should issue a message if the first character in the job name is not 'D'.

```
IF EPLJ-JOBNAME-C (1) NOT = 'D'
```

Assume that the third character in the job name must be 'P' or 'T'. The standard should issue a message if the third character in the job name is not 'P' or 'T'.

```
IF EPLJ-JOBNAME-C (3) NOT = ('P' AND 'T')
```

**Testing a Group of Characters in a Multi-Character Field**

A standard might apply to a group of characters within a multi-character field. For example, a site might require that the first two characters of a job name be one of several two character codes.

If you use the Prompter to write a Standards program or write it by copying CTRKIT, there are some predefined fields to make checking groups of characters easier. One of these is an eight character field called WORK8. WORK8 has been defined so that you can access the starting or ending characters in it using special fields named FIRST1, FIRST2, FIRST3, ..., FIRST7, and LAST1, LAST2, LAST3, ..., LAST7.

If you move a JCL parameter, such as the job name, program name, DD name, or one of the qualifiers in the dataset name, into WORK8, you can access the first character in it using FIRST1, the first two characters using FIRST2, the first three characters using FIRST3, and so on.

Similarly, you can access the eighth character using LAST1, the seventh and eighth characters using LAST2, and so on.

C  
o  
n  
t  
e  
n  
t  
s  
  
o  
f  
  
W  
O  
R  
K  
8

### Testing the First Two Characters of the Job Name

This standard issues a message if the first two characters of the job name are not PR:

```
MOVE EPLJ-JOBNAME TO WORK8
IF FIRST2 NOT = 'PR'
```

### Testing the First Three Characters of the Job Name

This standard issues a message if the first three characters of the job name are not PRD, PR2, or PR3:

```
MOVE EPLJ-JOBNAME TO WORK8
IF FIRST3 NOT = ('PRD' AND 'PR2' AND 'PR3')
```

## Testing a Group of Characters in a Field

It might be sometimes necessary to test characters in the middle of a field. For example, you might need to test to see if the third and fourth characters of the job name are appropriate. CTRKIT has a built in function, called SUBSTRING, that makes this very easy.

The SUBSTRING functions have the general form 'SUBSTRING-S-N' where 'S' is the position of the starting character and 'N' is the number of characters to extract. SUBSTRING-3-2 would extract two characters, beginning with the third character in the specified string. The extracted substring is placed in the field 'WORK8'.

The below figure shows how the SUBSTRING function starts with character and moves characters to the left (or first) part of WORK8.

S  
U  
B  
S  
T  
R  
I  
N  
G  
  
F  
u  
n  
c  
t  
i  
o  
n

You could also accomplish this directly:

```
MOVE HOLD8 (3) TO WORK8 (1).
MOVE HOLD8 (4) TO WORK8 (2).
```

However, you might find that the SUBSTRING function provided in CTRKIT is easier to use, especially if three or more characters need to be moved.

## Using the SUBSTRING Functions

JOB/SCAN provides these substring functions:

SUBSTRING-2-1	SUBSTRING-4-2
SUBSTRING-2-2	SUBSTRING-4-3
SUBSTRING-2-3	SUBSTRING-4-4
SUBSTRING-2-4	SUBSTRING-4-5
SUBSTRING-2-5	SUBSTRING-5-1
SUBSTRING-2-6	SUBSTRING-5-2
SUBSTRING-2-7	SUBSTRING-5-3
SUBSTRING-3-1	SUBSTRING-5-4
SUBSTRING-3-2	SUBSTRING-6-1
SUBSTRING-3-3	SUBSTRING-6-2
SUBSTRING-3-4	SUBSTRING-6-3

SUBSTRING-3-5                      SUBSTRING-7-1  
 SUBSTRING-3-6                      SUBSTRING-7-2  
 SUBSTRING-4-1

### Testing Third and Fourth Characters of Job Name

This example shows how to test the third and fourth characters of the job name. It issues a message if the third and fourth characters of the job name are not AB, CD, or EF. The SUBSTRING function expects the starting field to be in HOLD8; then places its output in WORK8. The FIRST2 accesses the first two characters in WORK8, as illustrated in figure: [SUBSTRING Function](#).

```
MOVE EPLJ-JOBNAME TO HOLD8.
PERFORM SUBSTRING-3-2.
IF FIRST2 NOT = ('AB' AND 'CD' AND 'EF')
```

## Testing Numeric and Character Fields

The fields shown in the previous examples are character fields. There are also numeric fields in the EPL. Numeric fields may only contain numbers. Arithmetic may only be performed on numeric fields. Character fields may contain characters or numbers, but do not support arithmetic. Fields that may contain either characters or numbers are called alphanumeric.

The definition of a field specifies that it is either character or numeric. View the CEPL member of the JOB/SCAN SJSSTDL library. If the PICTURE (or PIC) part of the definition begins 'X' (such as 'PIC X(8)'), it is an alphanumeric (character) field (the number in parentheses is the number of characters in the field). If it begins with a '9', it is an unsigned number; it can only be positive. If it begins with 'S9', it is a signed number; it can be either positive or negative. For additional information on PICTUREs, see [Adding Fields to WORKING-STORAGE](#).

### Testing a Field that Contains a Number

When testing a character field, put quotes around the comparison string, even if the string contains only numbers. For example:

```
IF EPLJ-CLASS = '3'
```

When testing a numeric field, do not put quotes around the number being tested. For example:

```
IF EPLJ-PRIORITY = 8
```

### Testing Whether a Character in the Job Name is Numeric

Sometimes you might need to check to see if a field contains a numeric character or alphabetic character. There is a special word (NUMERIC) in COBOL that facilitates this test. This test allows you to issue a message if the second character in the job name is not numeric:

```
IF EPLJ-JOBNAME-C (2) NOT NUMERIC
```

## Testing JCL Parameters

The test for whether a character field was specified in the JCL, is to compare it to the COBOL reserved word SPACES. The test for whether a numeric field was specified, use the COBOL reserved word ZERO.

### Testing Whether NOTIFY is Specified on the JOB Statement

This statement tests if NOTIFY is not specified on the JOB statement:

```
IF EPLJ-NOTIFY = SPACES
```

### Testing Whether DEST is Specified on the OUTPUT Statement

This statement tests if DEST is not specified on the OUTPUT statement:

```
IF EPLO-DEST = SPACES
```

### Testing Whether PRIORITY is Specified on the JOB Statement

This statement tests if PRIORITY is specified on the JOB statement:

```
IF EPLJ-PRIORITY NOT = ZERO
```

### Testing that SMS DATACLAS is Specified when Allocating New Datasets

These statements test if DATACLAS is not specified on the DD statement. Add them to the CALL-DD-ENTRY SECTION:

```
IF EPLD-DISP-1-AS-CODED = NEW
AND EPLD-DATACLAS = SPACES
```

### Testing BLKSIZE and LRECL Length

JOB/SCAN automatically tests the BLKSIZE against the LRECL length to determine if the BLKSIZE is appropriate for the LRECL length.

If you are running DFP3.2, the best way to optimize the BLKSIZE is to set it to zero.

You can use JOB/SCAN's JCL Change Facility to automatically set the BLKSIZE to zero.

If you just want to check the BLKSIZE and generate an error message use this text:

```
IF EPLD-BLKSIZE NOT = ZERO
```

### Testing the REGION Parameter

This statement tests if REGION is specified on the JOB statement:

```
IF EPLJ-REGION NOT = SPACE
```

This statement tests if REGION is specified on the EXEC statement:

```
IF EPLX-REGION-AS-CODED NOT = SPACE
```

### Testing the REGION for a Specific Value

This statement tests if REGION is not equal to 1024K:

```
IF EPLJ-REGION NOT = '1024K'
```

To check the value coded for the REGION parameter to see if it is greater than or less than a given value, it must first be converted to numeric. For an example of converting alphanumeric to numeric, refer to [Advanced Topics](#).

## Adding Fields to WORKING-STORAGE

All of the standards shown so far have used only pre-defined field names. However, you are not limited to using fields defined by ASG. You may define your own fields for holding parts of JCL parameters, doing calculations, or any other purpose. You define these fields in the WORKING-STORAGE SECTION of the Data Division in the Standards program.

The Standards Prompter cannot add user fields to working storage. They must be added to the Standards program using a text editor such as ISPF Edit.

There are many ways to define fields in WORKING-STORAGE. Only a few of the many options are necessary for writing Standards programs. These options are described below. For information on the other options, see the IBM manual *VS COBOL for OS/VS* (GC26-3857).

## Field Names

Field names may contain letters (A-Z), digits (0-9), and hyphens (-). They may contain up to 30 characters. They may not contain blanks.

Make the Standards program easier to understand, with field names that describe what the field is used for. It is also a good idea to add a prefix to field names to indicate where they are defined. For example, 'EPL' is used as a prefix for all of the fields defined in the Extended Parameter List. You might wish to use the prefix 'WS-' for your field names to indicate that they are defined in the working storage section. Following these guidelines, 'WS-DD-COUNT' is a more meaningful name than 'WORK-FIELD'.

## Level Numbers

WORKING-STORAGE fields are usually defined in groups. Level numbers establish the relationship between group names and field names. Fields with higher level numbers are part of groups with lower level numbers. In this example, EPLJ-JOB-DATA at level 05 is part of EXIT-PARAMETER-LIST at level 01, and EPLJ-JOBNAME at level 10 is part of EPLJ-JOB-DATA at level 05.

```
01 EXIT-PARAMETER-LIST.
05 EPLJ-JOB-DATA.
10 EPLJ-JOBNAME.
```

The level numbers do not have to be contiguous. In this example, they skip from 01 to 05 to 10. The sample Standards programs and Standards programs built by the Standards Prompter space the level numbers by 5 (01, 05, 10, 15, etc.). COBOL allows level numbers 01-49, 66, 77, and 88. Levels 66, 77, and 88 have special properties. Levels 66 and 77 are not needed to write Standards programs and are not covered in this documentation. Level 88 is described in [Using Named Conditions \(88-level fields\)](#).

The top level is always 01. You cannot define other levels without having an 01 level. There can be any number of 01 level items in WORKING-STORAGE. Level 01 definitions must begin in Area A (columns 8-11) but may extend into Area B. The other levels must begin in Area B (columns 12-80).



There are two kinds of names defined in WORKING-STORAGE: group names and elementary item names. Groups consist of elementary items. Each group definition has a period immediately following the name. Each elementary item must have a picture to indicate the size and type of the field before the period that ends the line.

When adding a new field to WORKING-STORAGE, take care not to add it into an existing group. That can cause the program to stop working properly. The safest place to add new fields is after all of the existing fields in WORKING-STORAGE.

In the example below WS-WORK-AREAS, and WS-WORK3 are Group Items, WS-WORK1, WS-WORK2, WS-WORK3A and WS-WORK3B are Elementary Items.

```
01WS-WORK-AREAS .
05WS-WORK1PICX (4) .
05WS-WORK2PICX (4) .
05WS-WORK3
10WS-WORK3APICX (2) .
10WS-WORK3BPICX (6) .
```

## Pictures

Pictures are used to define the size and type of data that a WORKING-STORAGE field may contain. There are several data types available to COBOL programs. Standards programs typically use only the alphanumeric (character) and numeric data types.

Alphanumeric fields are indicated by the letter 'X'. Numeric fields are indicated by a '9'. There are two ways to indicate the size of the field. One way is by the number of X's or 9's. Thus 'XXXX' indicates a four character field and '999' indicates a three digit numeric field. The other way is to put the size in parentheses after the X or 9. Thus 'X(8)' indicates an eight character field.

The picture clause may be written out in full after the field name, as in this example:

```
05WS-ACCT-FIELD-1PICTURE IS X(8) .
```

However, 'PICTURE IS' is usually abbreviated to PIC:

```
05WS-ACCT-FIELD-1PIC X(8) .
```

### Using Named Conditions (88-level fields)

Some JCL parameters have a limited set of valid values. To make these fields easier to test, a named condition (88-level field) is defined for each value.

For example, the EPLD-DISP-1 parameter can be 'N' (NEW), 'O' (OLD), 'S' (SHR), or 'M' (MOD). There is an 88-level field defined in CEPL for each of the possible values, as shown here:

```
15EPLD-DISP-1PIC X.88EPLD-NEWVALUE 'N' .
88EPLD-OLDVALUE 'O' .
88EPLD-MODVALUE 'M' .
88EPLD-SHRVALUE 'S' .
```

Instead of testing the value of EPLD-DISP-1, you can test the named condition.

### Testing Whether TYPRUN=HOLD is Specified

In this example, the standard issues a message if TYPRUN=HOLD is specified in a JOB EPLJ-TYPRUN statement. The TYPRUN parameter can be tested in two different ways. One way is to test the EPL field which contains the TYPRUN

parameter (EPLJ-TYPRUN). The other way is to test the 88-level field (EPLJ-TYPRUN-HOLD) which is set if TYPRUN=HOLD is specified.

Method 1. Test the value in the field EPLJ-TYPRUN. Add to the CALL-JOB SECTION:

```
IF EPLJ-TYPRUN = 'H'.
```

Method 2. Test the named condition EPLJ-TYPRUN-HOLD.

```
IF EPLJ-TYPRUN-HOLD
```

### Ensuring that RLSE is Specified on the SPACE Parameter

This standard issues a message if the RLSE option is not specified on the SPACE parameter on a DD statement. Add to the CALL-DD-ENTRY SECTION:

```
IF EPLD-SPACE-NOT-RLSE
```

### Testing RLSE on the SPACE Parameter

This standard issues a message if the RLSE option is specified on the SPACE parameter on a DD statement. Add to the CALL-DD-ENTRY SECTION:

```
IF EPLD-SPACE-IS-RLSE
```

## Testing Dataset Name Nodes

Standards can apply to a particular node of the dataset name. For example, a standard might require that the third node of the dataset name be 'CNTL', 'COBOL', 'CLIST', or 'LOAD'. To test a particular node of a dataset name, use the field name 'EPLD-DSN-NODE' followed by the node number in parentheses. These examples illustrate this concept.

### Testing the First Node of Dataset Name

This statement tests if the first node of the dataset name is not 'PROD':

```
IF EPLD-DSN-NODE (1) NOT = 'PROD'
```

### Testing the Third Node of Dataset Name

This statement tests if the third node of the dataset name is not 'CNTL', 'COBOL', 'CLIST', or 'LOAD':

```
IF EPLD-DSN-NODE (3) NOT = ('CNTL' AND 'COBOL' AND 'CLIST' AND 'LOAD')
```

### Testing the Number of Nodes in a Dataset Name

This statement tests if the dataset name has more than three nodes:

```
IF EPLD-DSN-NODE (4) NOT = SPACES
```

### Testing the Second Node of Dataset Name

This statement tests if DISP=NEW was specified and the second node of the dataset name is not 'PROD':

```
IF EPLD-NEW AND EPLD-DSN-NODE (2) NOT = 'PROD'
```

### Testing Characters 1 and 2 of Second Node

These statements test that, for new datasets (DISP=NEW), if the first two characters of the second node of the dataset name are not the same as the third and fourth character of the job name:

```

IF EPLD-DISP-1-AS CODED = NEW
MOVE EPLD-DSN-NODE (2) TO WORK8
IF WORK8-C (1) NOT = EPLJ-JOBNAME-C (3)
AND WORK8-C (2) NOT = EPLJ-JOBNAME-C (4)

```

## Using UNSTRING

The UNSTRING function splits a character string into substrings (smaller character strings). In JOB/SCAN Standards programs, it is most often used to test a job or step accounting field.

UNSTRING uses delimiter characters, such as commas and parentheses, to determine where to split the string. The delimiter characters are specified in UNSTRING's 'DELIMITED BY' option. UNSTRING places the substring in user-defined fields designated by the 'INTO' parameter. The delimiter characters are not included in the substrings. For example, if UNSTRING is coded specifying:

```

UNSTRING EPLD-DSN
DELIMITED BY '.' INTO WS-NODE1 WS-NODE2.

```

and is executed on the string 'SYS1.PROCLIB', WS-NODE1 would contain 'SYS1' and WS-NODE2 would contain 'PROCLIB'.

UNSTRING stops filling in a substring when it encounters a delimiter. So if the first character in the string is a delimiter, the first field specified in the INTO parameter is blank.

You need to make sure that the receiving work fields are defined to be long enough to accommodate the expected data strings.

### Testing the Second Accounting Field

Assume that the JOB statement has two or more accounting fields and that the standard should send a message if the second accounting field on the JOB statement is not 'AN01', 'AN02', or 'PN01'. Because the accounting field begins with a delimiter (a left parenthesis), the first string extracted by UNSTRING is blank.

Add these field definitions to WORKING-STORAGE:

```

01 WORK-ITEMS.
05 WS-BLANK                PIC X.
05 WS-ACCT-FIELD-1        PIC X(8).
05 WS-ACCT-FIELD-2        PIC X(4).

```

The line '01 WORK-ITEMS.' is already be in the program if the program was created using the Prompter or by copying CTRKIT. If it is not there, add it at the beginning of the WORKING-STORAGE. Add the 05-level fields after any other fields defined under WORK-ITEMS.

Add to the CALL-JOB SECTION:

```

UNSTRING EPLJ-ACCOUNTING-DATA
DELIMITED BY '(' OR ',' OR ')'
INTO WS-BLANK WS-ACCT-FIELD-1 WS-ACCT-FIELD-2.
IF WS-ACCT-FIELD-2 NOT = ('AN01' AND 'AN02' AND 'PN01')
. . .

```

## Counting Items

There are times when counting items of a certain type might be necessary. For example, you might have a standard that limits the maximum number of steps in a job to 10.

**To count these steps**

1. Define a signed-numeric WORKING-STORAGE field to serve as the counter.
2. Initialize the counter to zero.
3. Increment the counter using the ADD verb, or decrement using the SUBTRACT verb.

**Defining a Counter**

Set up a WORKING-STORAGE field similar to this example and initialize it to 0:

```
01WS-STEP-COUNT PIC S9(5) VALUE 0.
```

For additional information, see [Adding Fields to WORKING-STORAGE](#)

**ADD Verb**

The ADD verb is used to add two fields together. It has a number of formats that can be used. This section covers the easiest format here for keeping a counter.

Format:

```
ADD n TO data-item-1.
```

Example:

```
ADD 1 TO WS-STEP-COUNT.
```

**To implement this standard**

1. Add the WS-STEP-COUNT to the WS SECTION.
2. Add this statement to the CALL-JOB SECTION:

```
MOVE 0 TO WS-STEP-COUNT.
```

3. Add to the CALL-PROGRAM SECTION:

```
ADD 1 TO WS-STEP-COUNT.  
IF WS-STEP-COUNT > 10
```

As you can see, counting items can be used for other purposes, such as counting DDs, PROCs, etc.

For additional information on arithmetic, see [Advanced Topics](#).

**SUBTRACT Verb**

SUBTRACT has a format similar to ADD. When the counter is initialized with a starting value, SUBTRACT is used to decrement the counter.

Format:

```
SUBTRACT n FROM data-item-1.
```

Example:

```
SUBTRACT 1 FROM WS-STEP-COUNTER.
```

## Standards Requiring Calculations

Some standards require calculations that use a formula to calculate a value. For example, you might need to limit the maximum amount of space allocated to each disk file. To do this, you need to add the primary space amount to the secondary space amount. Remember that the secondary amount may use up to 15 extents. This means that you first multiply the secondary amount by 15 and add it to the primary amount.

### COMPUTE Verb

The COMPUTE verb is used to calculate a resulting field by adding, subtracting, multiplying, and dividing two or more fields or numbers.

Format:

```
COMPUTE data-item-1 = arithmetic expression.
```

Arithmetic expressions consist of one or more number fields, one or more numbers, and one or more of these operations: +, -, \*, /.

Example:

```
COMPUTE WS-MAX-DS-SIZE = EPLD-SPACE-PRIMARY +
(EPLD-SPACE-SECONDARY * 15)
```

All fields used in arithmetic statements must be defined as numeric (PIC 9..., or PIC S9...). In some cases, an explicit number (1, 2, 3, ...) may be used in place of a field name, such as 15 as shown in the previous example.

You can see that you can make any kind of calculations that are necessary using the COMPUTE verb. For additional arithmetic verbs, see [Advanced Topics](#).

All fields involved in a COMPUTE statement MUST be defined as numeric and must contain a numeric value. You may NOT divide by zero. This results in an abend SOCB when you execute the program. The result field must be defined large enough to hold the largest possible value.

### PERFORM Verb

The PERFORM verb can be used to:

- Invoke code that is common to more than one standard. For example:

```
PERFORM Z999-SET-ERROR.
```

This example statement invokes a routine provided with JOB/SCAN for the purpose of returning standards-detected errors back to JOB/SCAN for inclusion into the SJL.

- Segregate some detail statements from the main body of the logic of the standard. For example:

```
PERFORM 100-CHECK-REGION.
```

It is best to give the set of statements to be performed a descriptive name and make it a section as shown below:

```
100-CHECK-REGION SECTION.
    MOVE ...
    ADD ...
    IF ...
```

All statements following a SECTION statement are considered to be part of that section until another SECTION is encountered.



Do NOT embed a SECTION to be performed in the main body of the pre-defined SECTIONS of the standards program. Place the SECTIONS that you create at the end of the program.

## Using the CALL-\*SECTIONS

There are several different sections predefined in the sample Standards programs and in Standards programs built using the Standards Prompter. (See [Standards Program Sections](#)) There is a section for each type of JCL statement, for example, the CALL-DD-ENTRY SECTION for checking DD statements. Put standards that apply to a single type of JCL statement into the corresponding section of the Standards program.

Some standards apply to a group of JCL statements. These standards usually require code in a section of the Standards program that applies to a group of JCL statements. There are several of these sections. The CALL-INITIAL SECTION is called once at the beginning of JOB/SCAN execution. The CALL-JOB SECTION is called at the beginning of each job. The CALL-PROGRAM SECTION is called at the beginning of each step. The CALL-END-STEP SECTION is called at the end of each step. The CALL-END-JOB SECTION is called at the end of each job. The CALL-FINAL SECTION is called at the end of JOB/SCAN execution.

For example, a standard might require each step to have a SYSUDUMP DD statement. This standard cannot be checked by testing any single JCL statement. Each DD statement in the step must be checked and a message issued if none of them is a SYSUDUMP DD statement.

In a Standards program, this is done by putting code in the CALL-DD-ENTRY SECTION that tests each DD name and sets a flag if it finds SYSUDUMP. At the end of the step, code is added to the CALL- END-STEP SECTION to check the flag and issue a message if the SYSUDUMP flag is not set.

### Testing to Ensure that each Step has a SYSUDUMP DD Statement

#### *To check for the existence of SYSUDUMP DD*

1. Define a 1-character WORKING-STORAGE switch.
2. Set the switch when you encounter a SYSUDUMP in the CALL-DD-ENTRY SECTION.
3. Check the switch at the end of the step.

In this case, the cycle is: start of step, DD statements in the step, and end of step. Other cycles can be PROC, JOB, or the entire RUN.

4. Add these field definitions to WORKING-STORAGE.

```
01 WS-AREAS.
05 WS-SYSUDUMP-INDICATOR PIC X.
   88 WS-SYSUDUMP-FOUND VALUE 'Y'
   88 WS-SYSUDUMP-NOT-FOUND VALUE 'N'.
```

The line '01 WS-AREAS ' is already be in the program if the program was created using the Prompter or by copying CSMP. If it is not there, add it at the beginning of the WORKING-STORAGE (add the 05-level fields under WORK-ITEMS).

5. Add to the CALL-PROGRAM SECTION (beginning of step):

```
MOVE 'N' TO WS-SYSUDUMP-INDICATOR.
```

This sets the switch to the initial condition so if SYSUDUMP is later found, the program has a reliable test.

6. Add to the CALL-DD-ENTRY SECTION (as events occur):

```
IF EPLD-DDNAME = 'SYSUDUMP'  
MOVE 'Y' TO WS-SYSUDUMP-INDICATOR.
```

7. Add to the CALL-END-STEP SECTION (end of step):

```
IF WS-SYSUDUMP-NOT-FOUND.
```

## Testing for a JCLLIB Statement

### *To test for the presence of a JCLLIB statement:*

1. Add to WORKING-STORAGE:

```
01 WS-AREAS.  
05 WS-JCLLIB-INDICATOR          PIC X.  
   88 WS-JCLLIB-FOUND           VALUE 'Y'.  
   88 WS-JCLLIB-NOT-FOUND       VALUE 'N'.
```

2. Add to the CALL-JOB SECTION (beginning of job):

```
MOVE 'N' TO WS-JCLLIB-INDICATOR.
```

3. Add to the CALL-JCLLIB SECTION:

```
MOVE 'Y' TO WS-JCLLIB-INDICATOR.
```

4. Add to the CALL-END-JOB SECTION:

```
IF WS-JCLLIB-NOT-FOUND
```

### *To test for the absence of a JCLLIB Statement*

Because the JCLLIB statement can be included anywhere in the JOB, there is a special call section in the Standards program. If the program gets called by JOB/SCAN for a JCLLIB statement, there must have been a JCLLIB statement. Therefore, to generate an error message, these lines added to the CALL-JCLLIB SECTION are sufficient.

```
MOVE '201 JCLLIB STATEMENT NOT ALLOWED'  
   TO EPLC-ERROR-MESSAGE  
MOVE 08 TO EPLC-SEVERITY  
PERFORM Z999-SET-ERROR.
```



Many of the examples include the entire set of statements which generate the error message for completeness.

## Multiple IFs and ELSEs

A Standards program checking one standard for a particular requirement might need to perform additional checking if the requirement is not met. For example, different standards might apply to a DD statement for a new dataset than apply to a DD statement for an existing dataset.

The Standards program could handle this case with two groups of IF statements. The first might begin:

```
IF EPLD-DISP-1-AS-CODED = 'NEW'
    check standards for new data sets...
```

While the second begins:

```
IF EPLD-DISP-1-AS-CODED NOT = 'NEW'
    check standards for existing data sets...
```

Another approach is to combine the two groups of IF statements into one IF statement using the ELSE option.

```
IF EPLD-DISP-1-AS-CODED = 'NEW'
    check standards for new data sets...
ELSE
    check standards for existing data sets...
```

Both approaches achieve the same results. Use the one that is easier for you.

There might be other IF statements within the IF/ELSE group of statements, and some of them might also have ELSEs. Each ELSE is paired with the closest previous IF statement that is not already paired with an ELSE. So it is sometimes necessary to add an ELSE to an IF statement within an IF/ELSE group so that the ELSE is paired with the correct IF. The added ELSE statement is often:

```
ELSE NEXT SENTENCE
```

It is a good idea to indent each IF within the IF/ELSE group to show that it is subordinate to the previous IF. It is also a good idea to line up each ELSE under the corresponding IF to make it easier to see which IFs go with which ELSEs.

There should be no periods within the IF/ELSE group of statements; such a period terminates the IF prematurely. There should always be a period at the end of the IF/ELSE group of statements.

### Examples

These examples use different techniques to illustrate two mutually exclusive requirements: that new datasets must have a data class explicitly assigned; or the data class parameter for existing datasets is not necessary. The sentence in each example is ended by the period at the end of the PERFORM Z999-SET-ERROR statement. Both examples issue one message if DISP=NEW and the DATACLAS is not specified, and a different message if the DATACLAS is specified on an existing dataset.



### Example 1

The approach here twice tests the DISP parameter and the DATACLAS parameter. Add this sentence the CALL-DD-ENTRY SECTION:

---

```

IF EPLD-DISP-1-AS-CODED = 'NEW'
  IF EPLD-DATACLAS = SPACES
    MOVE '501 MISSING DATACLAS ON NEW DATA SET'
      TO EPLC-ERROR-MESSAGE
    MOVE 08 TO EPLC-SEVERITY
    PERFORM Z999-SET-ERROR.
IF EPLD-DISP-1-AS-CODED NOT = 'NEW'
  IF EPLD-DATACLAS NOT = SPACES
    MOVE '502 DATACLAS IGNORED ON EXISTING DATA SET'
      TO EPLC-ERROR-MESSAGE
    MOVE 04 TO EPLC-SEVERITY
    PERFORM Z999-SET-ERROR.

```

---

### Example 2

The approach here tests the DISP parameter one time, but still tests the DATACLAS parameter twice. This example converts the third IF in Example 1 to ELSE, and adds an ELSE NEXT SENTENCE statement to close the second IF. The balance of the code is unchanged. The ELSE NEXT SENTENCE skips the remainder of the sentence because the one part of the requirement is fulfilled: a new dataset has a data class. Add this sentence to the CALL-DD-ENTRY SECTION:

---

```

IF EPLD-DISP-1-AS-CODED = 'NEW'
  IF EPLD-DATACLAS = SPACES
    MOVE '501 MISSING DATACLAS ON NEW DATA SET'
      TO EPLC-ERROR-MESSAGE
    MOVE 08 TO EPLC-SEVERITY
    PERFORM Z999-SET-ERROR
  ELSE NEXT SENTENCE
ELSE
IF EPLD-DATACLAS NOT = SPACES
  MOVE '502 DATACLAS IGNORED ON EXISTING DATA SET'
    TO EPLC-ERROR-MESSAGE
  MOVE 04 TO EPLC-SEVERITY
  PERFORM Z999-SET-ERROR.

```

---

## Defining External Tables

Previous examples have shown how to compare a JCL parameter with a list of values. This approach can be inconvenient if the list is very long or if it changes frequently. If the list is long, the COBOL code will be hard to understand. Every time the list changes, the Standards program must be updated and recompiled.

Lists of this type can be defined as User Control Tables within JOB/SCAN. This enables you to maintain lists outside of the Standards Program. The below figure illustrates the relationship between tables and the Standards program.

## User Control Tables

JOB/SCAN automatically loads the contents of the user control table from JSOPT02 or JSOPT03 into a control block in CEPL at execution time. The Standards program can use the table processor (J06YTBL) to easily look up items of interest in the user control tables.

All user control table entries begin with a 'U' in column 6. The second character is a table identifier. In the sample table in JSOPT02, the second character is an 'A'. Other tables would have a different second character to distinguish them from the UA table. By using a different letter for the second character, you may have up to 26 distinct tables (UA through UZ). This is accomplished by forming a key that is made up of the table type and the specified item.

In the below figure, the item NNNN would not be found in the table therefore, the return code would be non-zero. For a different item, such as EEEE with a lookup key of UAEEEE, it will be found and the return code will be zero.

## Contents of Table

### ***To create and use a table***

1. Create the User Control Table data in member JSOPT02 or JSOPT03.
2. Define the table layout (PICTURE) as part of WORKING-STORAGE.
3. Establish table values including the table request and key length.
4. Perform CALL to the table processor.
5. Check the return code from the table process.

Benefits include eliminating the need to code long and complex IF statements, and changes to valid system IDs are made by editing the table, which is part of JSOPT02 or JSOPT03. The Standards program does not have to be modified or recompiled.

## Using An External Table to Test Characters 3-5 of the Job Name

In this example, the third through fifth characters of the job card must be one of these application codes: ACP, ACR, GLD, PAY, SLS, EXP, IRS, BKP.

Define the UA Control Table by putting these entries in JSOPT02.

### Portion of JSOPT02 showing testing of JOBNAME characters

```
CONTROL TABLES
UAACP
UAACR
UAGLD
UAPAY
UASLS
UAEXP
UAIRS
UABKP
```

Add this code to WORKING-STORAGE:

```
05 WS-SYSTEM-TABLE.
   10 FILLER                                PIC X(5) .
   10 WS-SYSID-TABLE-ID                     PIC X(2)  VALUE 'UA'.
   10 WS-SYSTEM-ID                          PIC X(3)  VALUE SPACES.
   10 FILLER                                PIC X(70) .
```

Add the lines of code as shown in the below figure to the CALL-JOB SECTION of the Standards.

### COBOL code for the CALL JOB SECTION of the standards program.

```
*
* THIS CODE SETS UP PARAMETERS FOR THE TABLE ACCESSOR
*
   MOVE  EPLJ-JOBNAME TO WORK8.
   MOVE  FIRST3 TO WS-SYSTEM-ID.
   MOVE  WS-SYSTEM-TABLE TO EPL-USER-RCT-DATA.
   MOVE  +2 TO EPLU-REQUEST.
   MOVE  +5 TO EPLU-KEYL.
*
* CALL TABLE ACCESSOR
*
   CALL  'J06YTBL'USING EPL-USER-RCT
        EPL-USER-RCT-DATA.
*
* TEST THE RETURN CODE
*
   IF    RETURN-CODE NOT = 0
       MOVE  '010 INVALID JOBNAME SYSTEM ID' TO
           EPLC-ERROR-MESSAGE
       MOVE  08 TO EPLC-SEVERITY
       PERFORM Z999-SET-ERROR.
*
```

## Using Dynamic Tables

The table processor routines of JOB/SCAN can also be used to set up tables in memory where certain items can be stored. The contents of the tables can then be queried to determine if error messages should be generated.

An example of using this dynamic table handling function might be to table all the STEP names to make sure that there are no duplicate STEP names in a job. The examples that follow show all the statements that are necessary to check for duplicate STEP names.

These examples assume that the sections described here are performed by statements coded in the CALL-PROGRAM and CALL-END-JOB SECTIONS in the Standards program. There is also a requirement to place a 1-byte switch in WORKING-STORAGE as described below.

This line goes in WORKING-STORAGE:

```
01 WS-ALL-DELETEDPIC X VALUE 'N'.
```

This line goes in the CALL-PROGRAM SECTION:

```
PERFORM 100-STEPNAMES.
```

This line goes in the CALL-END-JOB SECTION:

```
MOVE 'N' TO WS-ALL-DELETED.
PERFORM 300-DELETE-ENTRIES UNTIL WS-ALL-DELETED = 'Y'.
```

The CALL-PROGRAM and CALL-END-JOB sections described on this page should be placed at the very end of the program. The code in the below figure tables the STEP names and generates an error message when there is a duplicate.

#### Code to table the STEP names

```
100-STEPNAMES SECTION.
*
*SETUP THE KEY AND CALL THE TABLE PROCESSOR
*REQUEST CODE +1 MEANS ADD UNIQUE
*
MOVE SPACESTO EPL-USER-RCT-DATA.
MOVE EPLX-STEPNAME TO EPLU-KEY-VALUE.
MOVE 'US' TO EPLU-TABLE-ID.
MOVE +10 TO EPLU-KEYL.
MOVE +1 TO EPLU-REQUEST.
*
CALL 'J06YTBL' USING EPL-USER-RCT EPL-USER-RCT-DATA.
*
IF RETURN-CODE NOT = 0
MOVE '010 DUPLICATE STEPNAME'
TO EPLC-ERROR-MESSAGE
MOVE +8 TO EPLC-SEVERITY
PERFORM Z999-SET-ERROR.
*
100-EXIT
EXIT.
```

It is a good idea to clear the table in the CALL-END-JOB SECTION. To do this you need to put code in to read the table sequentially and delete each entry that is a 'US' entry. The code to delete an entry is +4.

**Code to delete STEP names**

```

300-DELETE-ENTRIES SECTION.
*
*CALL THE TABLE PROCESSOR TO DELETE STEP NAMES FROM
*THE US TABLE.
*REQUEST CODE +4 MEANS DELETE THE KEY THAT MATCHES
*
  MOVE SPACES TO EPL-USER-RCT-DATA.
  MOVE 'US' TO EPLU-TABLE-ID.
  MOVE +2 TO EPLU-KEYL.
  MOVE +4 TO EPLU-REQUEST.
*
  CALL 'J06YTBL' USING EPL-USER-RCT
                    EPL-USER-RCT-DATA.
*
  IF RETURN-CODE NOT = 0
  MOVE 'Y' TO WS-ALL-DELETED.
*
300-EXIT.
EXIT.

```

For additional functions that can be performed with the table processor, refer to EPL-USER-RUN-CONTROL-TABLES of CEPL and EPLDOC in the SJSSTDL library.

## Advanced Topics

This section contains useful techniques and sample code that can help implement more sophisticated standards.

### Conversion from Alphanumeric to Numeric

This section describes how to convert an alphanumeric field to numeric so that it can be used to test the value. The EPLJ-REGION value is used in this example. Other alphanumeric fields could also be handled in a similar fashion.

To check the value coded for the REGION parameter, you need to be able to strip off the last character, which should be a 'K' or an 'M', right justify the numeric value to a numeric work field, and then test the value against your standard.

Ch  
ec  
ki  
ng  
th  
e  
RE  
GI  
O  
N  
Pa  
ra  
m  
et  
er

W =

**To check the value of the REGION parameter**

1. Define some fields in WORKING-STORAGE.
2. Add code to the CALL-JOB SECTION.
3. Add code to the very end of the Standards program.

The statements for each section are described in [WORKING-STORAGE](#) and [CALL-JOB SECTION](#).

**WORKING-STORAGE**

The code below defines all the work fields, counters, and subscripts that are necessary for converting the alphanumeric EPLJ-REGION field to numeric and computing the expanded numeric value of the REGION parameter.

**Code for working storage**

```

*
01 WS-AREAS.
*
05 WS-COUNTER PIC 9(5) VALUE ZERO.
*
05 WS-REGION.
10 WS-REGION-C OCCURS 8 TIMES PIC X.
05 WS-REGION-N.
10 WS-REGION-N-COCCURS 8 TIMES PIC 9.
*
05 WS-SUB-X PIC 9.
05 WS-SUB-N PIC 9.
*
05 WS-REGION-FACTOR PIC 9(8) VALUE ZERO.
05 WS-REGION-AMOUNT PIC 9(18) VALUE ZERO.
05 WS-REGION-COMP PIC 9(8) VALUE ZERO.
*
* THE FIELD BELOW IS MAX REGION ALLOWED
* EDIT THE VALUE TO CONTAIN THE APPROPRIATE NUMBER IN TOTAL
* BYTES. EXAMPLE: 256K SHOULD BE ENTERED AS 00256000.
*                4M  SHOULD BE ENTERED AS 04000000.
*
05 WS-REGION-LIMIT PIC 9(8) VALUE 00256000.
*

```

**CALL-JOB SECTION**

The code in the below figure converts the alphanumeric REGION (such as 256K) to a pure numeric value (256000). Additionally, it issues an error message if the expanded numeric value is greater than 256000.

The statement PERFORM 100-CHK-REGION works similarly to the PERFORM SUBSTRING function.

**Code to convert a value from alphanumeric to numeric**

```

CALL-JOB SECTION.
*
* CHECK FOR REGION
*
  MOVE EPLJ-REGION TO WS-REGION.
*
* CHECK-FOR-M-OR-K.
*
  MOVE ZERO TO WS-COUNTER.
  INSPECT WS-REGION TALLYING WS-COUNTER
    FOR ALL 'M'.
  IF WS-COUNTER = 0
    MOVE 1000 TO WS-REGION FACTOR.
  ELSE
    MOVE 1000000 TO WS-REGION FACTOR.
*
* RIGHT JUSTIFY THE VALUE INTO A NUMERIC FIELD
*
  PERFORM 100-CHK-REGION.
*
* SET UP FOR THE CALCULATION
*
  MOVE WS-REGION-N TO WS-REGION-COMP.
*
  COMPUTE WS-REGION-AMOUNT = WS-REGION-COMP *
    WS-REGION-FACTOR.
*
  IF WS-REGION-AMOUNT > WS-REGION-LIMIT
    MOVE '010 REGION IS MORE THAN 256 K' TO
      EPLC-ERROR-MESSAGE
    MOVE 4 TO EPLC-SEVERITY
    PERFORM Z999-SET-ERROR.
*
CALL-JOB-EXIT.
EXIT.

```

**END OF STANDARDS PROGRAM**

This code moves the numeric portion of the EPLJ-REGION field that is left justified into a target numeric field right justified. The same code can be used to process the EPLX-REGION or EPLX-REGION-AS-CODED fields by moving them into the WS-REGION field prior to performing the 100-CHK-REGION SECTION.

**Code for end of Standards program**

```

*
100-CHK-REGION SECTION.
*
  MOVE ZEROS TO WS-REGION-N WS-REGION-AMOUNT WS-REGION-COMP.
  MOVE 8 TO WS-SUB-X.
  MOVE 9 TO WS-SUB-N.
*
  PERFORM 110-MOVE-CHARACTERS
    UNTIL WS-SUB-X = 0.
*
110-MOVE-CHARACTERS SECTION.
*
  IF WS-REGION-C(WS-SUB-X) NUMERIC
    SUBTRACT 1 FROM WS-SUB-N
    MOVE WS-REGION-C(WS-SUB-X) TO WS-REGION-N-C(WS-SUB-N) .
    SUBTRACT 1 FROM WS-SUB-X.

```

## Arithmetic

There are two basic ways to do arithmetic in COBOL. One way is to write out the operations using the COBOL verbs ADD, SUBTRACT, MULTIPLY, and DIVIDE, with options such as TO, FROM, BY, and GIVING.

Examples:

```
ADD WS-COUNT-1, WS-COUNT-2 GIVING WS-TOTAL
SUBTRACT WS-DISCOUNT FROM WS-PRICE
```

### ADDITION

There are two basic formats for the ADD statement:

Format 1:

```
ADD data-item-1 TO data-item-2
```

In this format, the value of data-item-1 is added to the value of data-item-2 and the result replaces the value in data-item-2.

Format 2:

```
ADD data-item-1, data-item-2 GIVING data-item-3
```

In this format, the value of data-item-1 is added to the value in data-item-2 and the result is placed in data-item-3.

### SUBTRACTION

There are two basic formats for the SUBTRACT statement:

Format 1:

```
SUBTRACT data-item-1 FROM data-item-2
```

In this format, the value of data-item-1 is subtracted from the value of data-item-2, and the result replaces the value in data-item-2.

Format 2:

```
SUBTRACT data-item-1 FROM data-item-2 GIVING data-item-3
```

In this format, the value of data-item-1 is subtracted from the value in data-item-2, and the result is placed in data-item-3.

### MULTIPLICATION

There are two basic formats for the MULTIPLY statement:

Format 1:

```
MULTIPLY data-item-1 BY data-item-2
```



In this format, the value of data-item-1 is multiplied by the value of data-item-2, and the result replaces the value in data-item-2. For example:

```
MOVE 2 INTO WS-WORK.
MULTIPLY 3 BY WS-WORK
```

In this example, 3 is multiplied by the value in WS-WORK, 2, and the answer, 6, is placed in WS-WORK.

Format 2:

```
MULTIPLY data-item-1 BY data-item-2 GIVING data-item-3
```

In this format, the value of data-item-1 is multiplied by the value in data-item-2, and the result is placed in data-item-3. For example:

```
MOVE 4 INTO WS-FIELD-1.
MOVE 2 INTO WS-FIELD-2.
MULTIPLY WS-FIELD-1
  BY WS-FIELD-2
  GIVING WS-ANSWER.
```

In this example, the value in WS-FIELD-1, 4, is multiplied by the value in WS-FIELD-2, 2, and the result, 8, is placed in WS-ANSWER.

## DIVISION

There are two basic formats for the DIVIDE statement:

Format 1:

```
DIVIDE data-item-1 INTO data-item-2
```

In this format, the value of data-item-2 is divided by the value of data-item-1, and the result replaces the value in data-item-2. For example:

```
MOVE 4 INTO WS-DATA.
DIVIDE 2 INTO WS-DATA.
```

In this example, the value in WS-DATA, 4, is divided by 2, and the result, 2, replaces the value in WS-DATA.

Format 2:

```
DIVIDE data-item-1 INTO data-item-2 GIVING data-item-3
DIVIDE data-item-1 BY data-item-2 GIVING data-item-3
```

In this format, DIVIDE works differently depending on whether INTO or BY is used. Using INTO in format 2, the value of data-item-2 is divided by the value in data-item-1, and the result is placed in data-item-3. Using BY in format 2, the value of data-item-1 is divided by the value in data-item-2, and the result is placed in data-item-3.

The option REMAINDER may be specified in format 2 DIVIDES, as in this example. REMAINDER designates the field in which the remainder of the division should be placed.

```
DIVIDE 5 BY 2
  GIVING WS-QUOTIENT
  REMAINDER WS-REMAINDER.
```

In this example, 5 is divided by 2. Because 2 goes into 5 two times, WS-QUOTIENT is set to 2. Because 2 does not go evenly into 5, the remainder, 1, is placed in WS-REMAINDER.

## Generating JCL Change Commands with the Standards Program

When using JOB/SCAN to perform modifications to JCL, such as conversions, changes required in the move-to-production process, etc., you might encounter a situation that requires detailed analysis that goes beyond the design objectives of the JOB/SCAN JCL Change Facility. In such cases, a Standards program can be used to perform the detailed analysis, generate JCL Change commands, and write the commands to a file. The resulting file can be used as input to the JCL Change Facility to perform the actual changes.

One such example is when you might have to count the number of tape datasets in the JCL and based on that number, you might need to go back to the JOB statement and make a change to the CLASS parameter.

The figure below illustrates the concept using the Standards Facility to feed JCL Change commands into the step that executes JOB/SCAN for the purpose of making changes.

Ge  
ne  
ra  
tin  
g  
JCL  
L  
Ch  
an  
ge  
Co  
m  
m  
an  
ds

Fig.

### ***To make the above concept work***

1. Define the output file where the JCL Change commands will be written.
2. Set up WORKING-STORAGE fields where the JCL Change commands will be built, and a counter to count tape files.
3. Put code in the appropriate sections of the Standards program to perform the necessary logic and write the output records.
4. Open the output file in the CALL-INITIAL SECTION.
5. Determine what changes need to be made.
6. Write the records that contain the JCL Change commands.
7. Close the output file in the CALL-FINAL SECTION.

### **ENVIRONMENT DIVISION**

These statements define the output file that DD (CHGS) uses to contain the change commands.

```

*
INPUT-OUTPUT SECTION.
*
FILE-CONTROL.
*
    SELECT CHGS-FILE ASSIGN TO UT-S-CHGS.

```

## DATA DIVISION

These statements defines the characteristics of the output file.

```

*
FILE SECTION.
*
FD CHGS-FILE
RECORDING MODE IS F
RECORD CONTAINS 80 CHARACTERS
BLOCK CONTAINS 0 RECORDS
LABEL RECORDS ARE STANDARD.
*
01 CHG-REC                PIC X(80) .

```

## WORKING-STORAGE SECTION

These statements define the record layout for the output change commands; it consists of the hard coded words and the two fields that contain the job name and the new class value:

```

01 WS-CHG-OUT-REC-1.
05 FILLER                PIC X(7)  VALUE 'IF JOB=' .
05 WS-CHG-JOBNAME        PIC X(8)  VALUE SPACES .
05 FILLER                PIC X(1)  VALUE SPACE .
05 FILLER                PIC X(7)  VALUE 'ON JOB ' .
05 FILLER                PIC X(15) VALUE 'CHG CLASS FROM ' .
05 FILLER                PIC X(5)  VALUE '* TO ' .
05 WS-CHG-NEW-CLASS      PIC X(1)  VALUE SPACE .
05 FILLER                PIC X(04) VALUE ' END' .
05 FILLER                PIC X(32) VALUE SPACES .

```

These statements define the counter field that keep track of tapes in a job.

```

01 WS-COUNTERS.
05 WS-TAPE-COUNTER      PIC 999 VALUE ZERO.

```

## CALL-INITIAL SECTION

This statement opens the output file.

```
OPEN OUTPUT CHGS-FILE.
```

## CALL-JOB SECTION

This statement resets the counter to zero, located at the beginning of a job.

```
MOVE 0 TO WS-TAPE-COUNTER.
```

## CALL-DD-ENTRY SECTION

These statements add 1 to the counter when a tape or cartridge dataset is present.

```
IF EPLD-TAPE OR EPLD-CARTRIDGE
ADD 1 TO WS-TAPE-COUNTER.
```

## CALL-END-JOB SECTION

These statements move the appropriate class value, based on the count, to the output field for the new class.

```

IF WS-TAPE-COUNTER > 0 AND < 3
  MOVE 'A' TO WS-CHG-NEW-CLASS.
*
IF WS-TAPE-COUNTER > 2 AND < 6
  MOVE 'B' TO WS-CHG-NEW-CLASS.
*
IF WS-TAPE-COUNTER > 5
  MOVE 'C' TO WS-CHG-NEW-CLASS.

```

These statements write out the change command if the counter is anything but zero, meaning that tapes are used in the job.

```

IF WS-TAPE-COUNTER > 0
  PERFORM 100-WRITE-CHG-COMMANDS.

```

## CALL-FINAL SECTION

This statement is used at the end of the run; the output file must be closed.

```

CLOSE CHGS-FILE.

```

## ADD TO THE END OF THE PROGRAM

These statements move the job name to the output area and writes the change commands.

```

*
100-WRITE-CHG-COMMANDS SECTION.
*
  MOVE EPLJ-JOBNAME          TO WS-CHG-JOBNAME.
*
  WRITE CHG-REC FROM WS-CHG-OUT-REC-1.
*
100-X.
  EXIT.

```

# COBOL Standards Testing and Debugging

Any computer process that is written by a human being might contain defects that must be found and corrected. JCL is one example of a computer process that is written and needs to be tested and debugged prior to being used in production. JOB/SCAN is the tool that helps with that process.

Similarly your standards, whether written using the Prompter or by using ISPF Edit, must be tested (and corrected when problems are found).

This section provides useful information about testing and debugging your standards.

- [Finding and Correcting Compiler Errors](#)
- [Setting Up Test JCL Scenarios](#)
- [Values Passed to Standards](#)
- [Section or Statement Execution](#)

## Finding and Correcting Compiler Errors

When there are errors in COBOL code, the COBOL compiler produces error messages. Sometimes it is hard to tell from the error message exactly what is wrong with the COBOL code. This section explains some of the common compiler error messages and gives some hints on how to find errors.

### Common Mistakes

You might be able to find an error by inspecting the program. These are some common kinds of mistakes to look for:

- Misspelled field names.
- Missing or misplaced periods.
- Incorrect placement on the line (such as a statement beginning in Area A, or an asterisk not in column 7 on a comment line).

### Reviewing Compiler Output and Locating Errors

Sometimes when the COBOL compiler finds an error, it produces an error message not only for the statement in error, but also for many other statements. In general, if you get many error messages, fix the first one and see if that solves any of the other problems.

It can be hard to tell where the error is in a line of COBOL, especially if it contains a long statement. The statement can be split across several lines to pinpoint the problem.

Sometimes the error is not on the line flagged by the COBOL compiler, but is on a previous line. This can happen, for example, if a period was omitted from a sentence. The sentence missing a period is often not flagged while later sentences are.

If you cannot find an error on the line flagged by the compiler, try making it and any other lines that form part of the same sentence, into comments by putting an asterisk (\*) in column 7. If you now get error messages on the lines following the ones commented out, the real error must be in a previous line.

In order to find out whether the standards compiled error free, you need to check the return code from the compile step and the link step in the job. If both steps executed and the link step completed with a return code of 00, then the program was compile-time error free.

If the link step did not execute, then the compile step probably produced a return code that is higher than 04. In this case you need to find the statement(s) that contain the error(s).

The COBOL program listing produced by the compiler is written to the SYSPRINT DD statement. Using SDSF or a similar utility, browse the SYSPRINT DD output file from the compile step. Page down to the end of the listing where the compiler error messages are displayed.

The error messages are listed in order of occurrence. Scroll to the first message. The sequence number along the left margin is the number that identifies each statement in your program. For example:

```
82 IKF1043I-W END OF SENTENCE SHOULD PRECEDE 05. ASSUMED PRESENT.
```

To find the statement at line 82, use this command:

```
F 82** FIRST
```

This command displays the line that the compiler thinks the error is on. For example:

```
081** 01 WS-WORK-REC
082** 05 WS-WORK-TYPE PIC X(2) .
083** 05 WS-WORK-USERID PIC X(8) .
```

When you look at line 82, you can see that there is nothing obviously wrong with it. However, the statement immediately preceding it, on line 81, is missing the period at the end. This is because the missing period cannot be detected by the compiler until it sees the next line.

Because this is a fairly common situation with COBOL error messages, it is advisable to always subtract 5 from the line number that the error message points to using the Find command. In this case you might have used the F 77\*\* FIRST. This way you have a better chance of seeing the actual statement causing the error immediately.

The ISPF FIND command shown here using the asterisks (\*\*) works only if the COBOL source is unnumbered in the edit member. This is the easiest way to find statements by using the line number in the error message.

You are now ready to edit your program to correct any errors. After all of the errors have been corrected, you can then re-compile.

## Setting Up Test JCL Scenarios

The most effective method for testing any particular standard that you have written is to create a small test JCL member that has just enough JCL in it to allow the two possible cases (true and false) to be tested in one execution.

For example, if you are testing the SPACE parameter on a DD statement that must have RLSE coded, then you can code two DD statements: DD1 and DD2. One statement with and one without the RLSE parameter can be run using your standards and in one test run, you can see the results from the two possible scenarios.

## Values Passed to Standards

Sometimes you might get test results that differ from what you expected. In these cases, it would be very useful to see what values JOB/SCAN passed to your standards in the EPL fields. JOB/SCAN has a utility called JSVERIFY that you can run to display the contents of all the fields in a given JCL statement. The utility can be run in either batch or on-line.

### How JSVERIFY Works

JSVERIFY is a two-step job that invokes JOB/SCAN on sample JCL. Using facilities provided by ASG, JSVERIFY displays:

- Each 80-byte JCL record.
- For each statement, every relevant Extended Parameter List field name, and the value that JOB/SCAN placed in it.

This eliminates the need to code DISPLAY statements for the Extended Parameter List fields within the standards program.

### Using JSVERIFY

Use the sample JCL in the SJSSTDL library. It executes JOB/SCAN, invoking a special Standards Program called JSVFYPM. JSVFYPM takes up to 20 run-time parameters (the use of which JSVERIFY describes) you use to specify what statement types you are interested in, such as JOB, DD, etc. The output of this run is a Structured JCL Listing of your test JCL followed by a detailed listing of all the fields and their values for the selected statement types. ASG recommends turning on only the statement types that you need so the output is minimized.

#### **To run JSVERIFY using the JOB/SCAN Execution Panel**

1. Edit the JSVERIFY member in the SJSSTDL library and modify it according to the directions contained within the member.
2. Specify the name JSVFYPM in the Standards Name field and the appropriate JCL statement type name in the parameter field. You also need to specify:

```
STANDARDS EXIT=JSVFYPM
```

3. Run JOB/SCAN using the F command.  
The detailed contents of the EPL fields will be displayed to the DD name SYSOUT which should normally be allocated to your terminal.

#### **To run JSVERIFY in batch**

1. Write sample JCL that contains the desired parameters that need to be tested. For example, to see what JOB/SCAN passes when a region parameter is coded on an EXEC statement, include this step in the sample JCL:

```
//STEP1 EXEC PGM=IEFBR14,REGION=1024K
```

2. Edit the JSVERIFY member in the SJSSTDL library and modify it according to the directions contained within the member.

JSVERIFY is a JOB/SCAN batch execution member with `STANDARDS EXIT=JSVFYPGM` coded in the INPUT SELECTION section of the SYSIN data stream and `EXIT NAME='JSVFYPGM, parameters'` coded in the RUN OPTIONS section of the SYSIN data stream. The parameters for JSVFYPGM consists of two types.

- JCL Statements – CMT, DD, EXPG, EXPR, IF, INCL, JES, JLIB, JOB, OUT, PROC, XMIT, SET
- Other parameters – CALL, CC, CC03, CNTRL, ERR, CMD

These parameters are described in the JSVERIFY member in the SJSSTDL library.

For example, to see all JOB and DD statement values and any associated JOB/SCAN messages, this would be the correct coding:

```
EXIT NAME =JSVFYPGM, 'JOB,DD,ERR'
```



## JSVERIFY

```

//JSVERIFY JOB (ACCT), 'PROGRAMMER NAME'
//*****
//*   DISPLAY VALUES PASSED TO THE JOB/SCAN STANDARDS EXIT (UTILITY TO
//*   ASSIST WITH STANDARDS EXIT DEBUGGING)
//*   REF:   STDLIB(JSVERIFY)
//*
//*   NOTE: MODIFY JOB STATEMENT AND ITEMS INDICATED BY ARROWS <==
//*   READ AND FOLLOW DIRECTIONS UNDER "REPORT HEADING" BELOW.
//*****
//*
//***** RUN JOB/SCAN *****
//JOBSCAN EXEC JS6PROC,PRM=S,
//          DJSNAME='ASG.PREFIX'           <== ASG PREFIX
//SYSOUT   DD DSN=&&SYSOUT,DISP=(MOD,PASS),
//          SPACE=(TRK,(5,1),RLSE),UNIT=SYSDA, <== VALID UNIT
//          DCB=(RECFM=FB,LRECL=121,BLKSIZE=1210)
//SYSIN    DD *
RUN OPTIONS
  REPORT HEADING ='STDS EXIT FIELD DISPLAY'
*
*   SELECT JSVFYPGM PARAMETER(S) .
*
* BLANK      - SPECIFIES ALL JSVFYPGM PARAMETERS EXCEPT CC03.
* CALL       - SPECIFIES ALL JSVFYPGM PARAMETERS EXCEPT CC03.
* CC         - DISPLAYS ALL EPL INFORMATION FOR AN INPUT CONTROL CARD
*             80-BYTE CARD IMAGE.
* CC03       - SETS THE EPLC-CC-2003 FIELD TO 'Y' IN JSVFYPGM.
* CMD        - DISPLAYS ALL EPL DATA ASSOCIATED WITH AN OPERATOR
*             COMMAND EMBEDDED IN THE JCL STREAM.
* CMT        - DISPLAYS ALL EPL INFORMATION FOR A JCL COMMENT STATEMENT.
* DD         - DISPLAYS ALL EPL INFORMATION FOR A DD JCL STATEMENT.
* DD,CNTRL   - DISPLAYS ALL CONTROL CARD 80-BYTE CARD IMAGES IN THE
*             EPL-CONTROL-STATEMENT TABLE WHEN PROCESSING AN
*             EPLC-CALL-DD-ENTRY TYPE CALL.
* ERR        - DISPLAYS ALL ERROR MESSAGES IN THE EPL-JOB-SCAN-ERRORS
*             TABLE FOR EACH JCL STATEMENT TYPE CALL TO THE STANDARDS
*             PROGRAM.
* EXPG       - DISPLAYS ALL EPL INFORMATION FOR AN EXEC PGM= STATEMENT.
* EXPR       - DISPLAYS ALL EPL INFORMATION FOR AN EXEC PROC= STATEMENT.
* IF         - DISPLAYS ALL EPL INFORMATION FOR IF/THEN/ELSE/ENDIF STMTS.
* INCL       - DISPLAYS ALL EPL INFORMATION FOR AN INCLUDE JCL STATEMENT.
* JES        - DISPLAYS ALL EPL INFORMATION FOR A JES2 OR A JES3 STMT.
* JLIB       - DISPLAYS ALL EPL INFORMATION FOR AN JCLLIB JCL STATEMENT.
* JOB        - DISPLAYS ALL EPL INFORMATION FOR A JOB JCL STATEMENT.
* OUT        - DISPLAYS ALL EPL INFORMATION FOR AN OUTPUT JCL STATEMENT.
* PROC       - DISPLAYS ALL EPL INFORMATION FOR A PROC JCL STATEMENT.
* SET        - DISPLAYS ALL EPL INFORMATION FOR A SET JCL STATEMENT.
* XMIT       - DISPLAYS ALL EPL INFORMATION FOR A XMIT JCL STATEMENT.
*
*   EXAMPLE - TO SEE ALL JOB AND DD STATEMENT VALUES AND ANY
*             ASSOCIATED ERROR MESSAGES, CODE:
*             EXIT NAME =JSVFYPGM, 'JOB,DD,ERR'
*
*   =====> MAXIMUM LENGTH OF JSVFYPGM PARAMETER IS 32 BYTES <=====
*   =====> '-----1-----2-----3--' <=====
EXIT NAME =JSVFYPGM, 'CALL,CC03'           <== SPECIFY TYPE
REPORT OPTIONS
  OPTIONS IN EFFECT      =PRINT           <== PRINT OPE REPORT?
  STRUCTURED JCL LISTING=PRINT           <== PRINT SJL REPORT?
INPUT SELECTION
  SELECTION EXIT =JSVFYSEL

  DSN=YOUR.JCLLIB           <== INPUT JCL LIBRARY
  SELECT=YOURJOB           <== JCL MEMBER
//*
//***** PRINT FORMATTED OUTPUT *****

```

```
//PRINTOUT EXEC PGM=IEBPTPCH
//SYSPRINT DD DUMMY
//SYSUT1 DD DSN= &&SYSOUT, DISP=(OLD,DELETE)
//SYSUT2 DD SYSOUT=*
//SYSIN DD *
PRINT MAXFLDS=1
RECORD FIELD=(120)
```



Make sure the SELECT= statement points to the sample JCL coded in Step 1. In addition, make sure the STANDARDS NAME= statement contains the desired statement type within quotations. THIS IS NOW EXIT NAME!!!!!!!!!!

3. Submit JSVERIFY and view the output.

The below figure shows output created by JSVERIFY for a sample Job Statement.

#### JSVERIFY Output for a Sample Job Statement

```
//PJGL0076 JOB (P07A,P1,6L),'D137 - SMITH', **GENERAL LEDGER MONTHLY**
// CLASS=P,MSGCLASS=S,MSGLEVEL=(1,1),
// TIME=1440,NOTIFY=D137BTS,USER=PROD,LINES=100

*** CALL-JOB SECTION *** EPLC-SYSTEM-NO = EPLC-JOB-INDEX = 00001 EPLC-JOBSTEP-INDEX = 000
EPLC-PROCSTEP-INDEX = 000 EPLC-FILE-INDEX = 000 EPLC-CALL-TYPE = J
EPLC-JS-MEMBER = PJGL0076 EPLC-JS-DSNAME = DSSRAW.WORK.CNTL
EPLJ-ERROR-INDICATOR = N 88 LEVEL VALUES ARE: Y OR N.
EPLJ-JOBNAME = PJGL0076
EPLJ-ACCOUNTING-DATA = (P07A,P1,6L)

EPLJ-ADDRSPC = 00 LEVEL VALUES ARE: V,R.
EPLJ-BYTES-COUNT = 000000
EPLJ-BYTES-ACTION = 88 LEVEL VALUES ARE: W,C,D.
EPLJ-CARDS-COUNT = 000000
EPLJ-CARDS-ACTION = 88 LEVEL VALUES ARE: W,C,D.
EPLJ-CLASS = P
EPLJ-COMMENT = **GENERAL LEDGER MONTHLY**
EPLJ-COMMENT-SOURCE = J 88 LEVEL VALUES ARE: D,F,I.
EPLJ-COND =
EPLJ-GROUP =
EPLJ-LINES-COUNT = 000100
EPLJ-LINES-ACTION = 88 LEVEL VALUES ARE: W,C,D.
EPLJ-JES-PRIORITY = 0000
EPLJ-MSGCLASS = S
EPLJ-MSGLEVEL = (1,1)
EPLJ-NOTIFY = D137BTS
EPLJ-PAGES-COUNT = 000000
EPLJ-PAGES-ACTION = 88 LEVEL VALUES ARE: W,C,D.
EPLJ-PASSWORD =
EPLJ-PERFORM = 0000
EPLJ-PRIORITY = 0000
EPLJ-PROGRAMMER-NAME = D137 - SMITH
EPLJ-RD =
EPLJ-REGION =
EPLJ-RESTART = N 88 LEVEL VALUES ARE: Y,N.
EPLJ-RESTART-VALUE-AS-CODED =
EPLJ-SECLABEL =
EPLJ-TIME-MINUTES = 001440
EPLJ-TIME-SECONDS = 00
EPLJ-TYPRUN = 88 LEVEL VALUES ARE: H,J,S,C.
EPLJ-USER = PROD
EPLJ-USER-DATA-MBR = PJGL0076
```

## Section or Statement Execution

There might be situations when an expected error message might not be produced and after using the techniques described above, there is still no obvious reason for the problem. In such cases, you need to see if the part of the

program that produces that error message is being executed. Often there might be a logic error in the program that might cause the section that produces that particular error message not to be executed at all.

To determine if this might be the case, you can use the DISPLAY verb. The output is displayed in the SYSOUT DD statement. Use this format for the format for the DISPLAY verb:

```
DISPLAY      'some literal'.
```

Example:

```
DISPLAY      'CHECKING DDNAME THIRD CHARACTER'.
```

By placing this statement just in front of the IF test in question, you can determine if this section of the program is being executed.

If the phrase 'CHECKING DDNAME THIRD CHARACTER' does not display on the SYSOUT when you test the program, there might be some other statements in the program causing that section not to be executed. If the phrase is displayed when you test, then the problem might be data related and not a logic error. You may place additional EXHIBIT NAMED statements in front of the IF test to display the exact contents of the fields that you are testing.

# REXX Standards

This topic gives an overview of using REXX to write standards.

- [Overview of REXX Standards Programs](#)
- [Additional REXX Functions](#)
- [REXX Documentation](#)

## Overview of REXX Standards Programs

A standards program written in the REXX language (herein called REXX standards) interacts with JOB/SCAN in the same manner as standards programs written in compiled languages such as COBOL or in Assembler. REXX standards may be used in any of the JOB/SCAN execution environments: foreground (ISPF panels), Edit macros, or batch jobs. REXX standards execute identically in all environments.

JOB/SCAN adds functions to the REXX environment to provide access to its data areas that are not otherwise accessible to a REXX exec. Specifically, REXX standards can access the Exit Parameter List (EPL), the Ux User Control Tables, the Control Card Table, and the JOB/SCAN Error Message Table.

The sample REXX standards provided with JOB/SCAN includes all the basic hand-shake routines. All that is needed are site-specific IF tests and error messages. [REXX Standards Implementation](#) contains detailed examples on how standards are implemented in REXX.

## Additional REXX Functions

JOB/SCAN makes five additional functions available to REXX standards. These functions provide access to JOB/SCAN data areas and tables that contain key information required by REXX standards. These function are listed in this table:

Function	Description
<a href="#">JSGET</a>	Fetch the value of an EPL field or global variable.
<a href="#">JSPUT</a>	Update the value of an EPL field or save a global variable.
<a href="#">JSMSG</a>	Write a user-generated message to the JOB/SCAN Structured JCL Listing or control the display of JOB/SCAN-generated messages.
<a href="#">JSTABLE</a>	Load a JOB/SCAN table into local REXX variables.
<a href="#">JSDUMP</a>	Diagnostic tool for displaying EPL field values and global variable values.

## REXX Documentation

Refer to the manuals listed in this table for more information on learning, writing, and debugging REXX standards:

Manual Title	Description
TSO/E V2 REXX/MVS User's Guide	If you are unfamiliar with REXX, start with this guide. Concentrate on the “Learning the REXX Language” and “Using REXX” sections.
TSO/E V2 REXX/MVS Reference	This manual is <i>the</i> source of REXX programming information. It contains detailed descriptions of the various REXX statements and built-in functions.
TSO/E V2 Messages	This manual contains a complete listing of REXX error messages. These messages have the format IRXaaaab.

---

# REXX Standards Implementation

This topic describes the use of REXX in writing standards for JOB/SCAN.

- [Setting up the REXX Environment with Required DDnames](#)
- [Specifying the Name of REXX Standards](#)
- [Writing a REXX Standard](#)
- [Sample Standards Statements](#)
- [Table of JCL Call Types](#)
- [Using Data Elements from the EPL](#)
- [EPL Field Examples](#)
- [Table Processing Overview and Procedure](#)

## Setting up the REXX Environment with Required DDnames

Allocate the DDnames in this table prior to using REXX standards:

DDname	Description
SYSEXEC	Allocate this ddname to the partitioned dataset containing REXX standards. JOB/SCAN first loads execs from DD=SYSEXEC; if it does not find the exec there, it looks in DD=SYSPROC. It may be allocated using File Allocations statements.
SYSTSPRT	Output from the REXX <i>SAY</i> statement, REXX error messages, and other tracing output are directed to this DDname. Normally SYSTSPRT is allocated to the terminal when running in the foreground environment. It may <i>not</i> be allocated using File Allocations statements.

The SYSEXEC or SYSPROC ddname must be allocated to the partitioned dataset (PDS) where the REXX standards program is to reside. Create a new PDS library, or use an existing SYSEXEC or SYSPROC dataset. Use the TSO ALLOC command, the FILE ALLOCATION statement in JSOPT02 or the batch DD statement to allocate the SYSEXEC or SYSPROC DDname to this PDS. Do not allocate the same library where the JOB/SCAN CLISTs (JEM, JSUB, and so on) reside to the SYSEXEC DDname. RKIT is the skeleton code for REXX standards. Make a copy of RKIT from the SJJSTDL dataset into a PDS library allocated to SYSEXEC or SYSPROC. The member name and REXX standards program code may be changed to meet site requirements. This documentation hereafter refers to this member as RKIT.

## Specifying the Name of REXX Standards

Specify the name of REXX standards in the same ways as a non-REXX standards: give JOB/SCAN the name of a standards program and the library in which it resides. JOB/SCAN determines whether it is a load module or a REXX exec.

JOB/SCAN searches for the named program in this order:

- Load module libraries allocated to JLLD.
- Datasets allocated to STEPLIB and/or the system link list.
- PDSs allocated to SYSEXEC.
- PDSs allocated to SYSPROC.

If the program is not located in the allocated libraries, JOB/SCAN generates this error message:

```
DSS0801E - STANDARDS PROGRAM <program name> IS NOT ACCESSIBLE
```



JOB/SCAN does not support the TSO/E-standard use of a percent sign (%) as a prefix character for the REXX exec name.

## Writing a REXX Standard

This section describes the major tasks in implementing requirements using the REXX language.

### To write a REXX standard

1. Identify a site standard. For example:

Priority (PRTY) may not be coded in the JOB statement.

2. Identify the field to test. For example, the field named EPLJ\_PRIORITY contains the priority value. Refer to [Using Data Elements from the EPL](#) for additional information.
3. Identify where the test should occur. For example:

Job statement tests should be put in the PROCESS\_JOB internal subroutine of the RKIT exec copy. When CALL\_TYPE is equal to 'J' the PROCESS\_JOB internal subroutine is called by the main line code in RKIT. The concept of call type and additional information can also be found in [CALL\\_TYPE Processing Subroutines](#).

4. Write the validation statement. For example:

```
epljob_priority = JSGET("EPLJ_PRIORITY")
if epljob_priority \= '0' then . . .
```

Refer to [JSGET](#) for information on the JSGET external function.

5. If the standards program finds a violation put the error message in a variable named msg\_text. For example:

```
msg_text = "070 PRIORITY PARAMETER NOT ALLOWED "
```



The first three digits of the message become the last three digits in the error message number (in this example, DSS9070).

6. Decide on the severity level (advisory, warning, or error) and message level (short, or extended text) of the message. Set fields MSG\_SEVERITY and MSG\_LEVEL to the appropriate values. For example:

```
msg_text = "070 PRIORITY PARAMETER NOT ALLOWED"
msg_severity = 8
```

```
msg_level = " "
```

Refer to the [Standards Error Messages](#) for information on severity and message levels.

7. Use the JSMSG external function to add the error message to an error table, which is returned to JOB/SCAN and is included automatically on the Structured JCL Listing. For example:

```
msg_text = "070 PRIORITY PARAMETER NOT ALLOWED"
msg_severity = 8
msg_level = " "
call JSMSG "ADD"
```

Refer to [JSMSG](#) for information.

## Sample Standards Statements

If a site standard requires all CLASS parameters to be “A”, code the following in the Process\_JOB internal subroutine within a copy of RKIT:

```
eplj_class = JSGET("EPLJ_CLASS")
if eplj_class \= "A" then
do
msg_text = "001 INVALID CLASS PARAMETER"
msg_severity = 8
msg_level = " "
call JSMSG "ADD"
end
```



If msg\_text, msg\_severity, or msg\_level are not defined prior to the call to JSMSG a DSS6305, JOB/SCAN generates the message “JMSG fetch of REXX variables failed”.

If the first character of all production job names must begin with “P”, code the following in the Process\_JOB internal subroutine within a copy of RKIT:

```
eplj_jobname = JSGET("eplj_jobname")
if SUBSTR(EPLJ_JOBNAME,1,1) \= "P" then
do
msg_text = "010 NOT A PRODUCTION JOB *****"
msg_severity = 4
msg_level = " "
call JSMSG "ADD"
end
```

If a site standard requires that a NOTIFY parameter must be included on the Job statement, code the following in the Process\_JOB internal subroutine within a copy of RKIT:

```
eplj_notify = JSGET("EPLJ_NOTIFY")
if eplj_notify = " " then
do
msg_text = "517 MISSING NOTIFY PARAMETER"
msg_severity = 4
msg_level = " "
call JSMSG "ADD"
end
```



## Table of JCL Call Types

The Standards facility processes several call types. Code designed to perform site-specific validation of the JCL statement parameters should be added to the appropriate call type subroutines. This table describes each call type and relates it to the applicable Standards EXEC Internal Subroutine:

Call Type	Code	Standards Exec Internal Subroutine	Description
BACK USER ERROR	H		Occurs after JOB/SCAN has processed fifty error messages. This feature is not available for standards programs implemented in REXX.
COMMENT GROUP END	G	Process_Comment_Group	Occurs at the end of a comment statement group. Before call is executed the EPL_COMMENT_CARD_DATA field is set to spaces.
COMMENT STATEMENTS	C	Process_Comment_Card	Occurs after a <code>/**</code> comment statement is processed. It may be used to verify the contents of <code>/**</code> comment statements.
CONTROL CARDS	L	Process_Control_Card	Occurs for each control card and may be used to validate the contents of the control cards.
DD STATEMENT	D	Process_DD	Occurs after each DD statement is processed and may be used to validate naming conventions for ddnames, dataset names, and unit names. You can also check VOLSERS, DCB information, SPACE allocations, SYSOUT parameters, DISP, DSORG, refer backs, and so on.
ENDIF STATEMENT	E	Process_ENDIF	Occurs after an IF/THEN/ELSE statement group is processed. At this point, validation of the IF/THEN/ELSE statement group is complete.
EXEC PGM STATEMENT	X	Process_EXEC_PGM	Occurs after an EXEC PGM= statement is processed and may be used to verify step names, program names, region parameters, and so on.
EXEC PROC STATEMENT	Q	Process_EXEC_PROC	Occurs after an EXEC PROC= statement is processed and may be used to validate any parameters specified on the EXEC PROC= statement. At this point, JOB/SCAN has stored only the symbolics specified on the EXEC PROC= statement; symbolics specified on the PROC PROTO TYPE statement are available at Process_PROC time.

Call Type	Code	Standards Exec Internal Subroutine	Description
IF/THEN START	F	Process_IF_THEN	Occurs after an IF statement is processed. At this point, IF processing is not complete.
INCLUDE END	U	Process_End_INCLUDE	Occurs after JOB/SCAN has processed all statements for the current INCLUDE.
INCLUDE START	N	Process_INCLUDE	Occurs after an INCLUDE statement is processed. It may be used to verify the member name specified in the INCLUDE statement.
INITIALIZATION	I	Process_Initial	Always called when JOB/SCAN is initialized and before any JCL is validated. Use this pass to set values for subsequent calls and to initialize any special switches used by these subsequent calls.
JCLLIB STATEMENT	B	Process_JCLLIB	Occurs after a JCLLIB statement is processed.
JES2 STATEMENT	2	Process_JES2	Occurs after a JES2 statement is processed. It may be used to validate parameters and options specified on the JES2 statement.
JES3 STATEMENT	3	Process_JES3	Occurs after a JES3 statement is processed. It may be used to validate parameters and options specified on the JES3 statement.
JOB END	K	Process_End_Job	Occurs after all JCL statements for the current job are processed. Any necessary switches required by the next job can be reset in this subroutine.
JOB START	J	Process_JOB	Occurs after the Job statement is processed and may be used to validate Job statement values, Job naming conventions, and so on.
OUTPUT STATEMENT	O	Process_OUTPUT	Occurs after an OUTPUT JCL statement is processed and may be used to verify standards on this statement.
PROC END	Q	Process_End_PROC	Occurs after all JCL statements for the current PROC are processed and may be used to verify programs executed by the PROC. At this point, the validation of the PROC is

Call Type	Code	Standards Exec Internal Subroutine	Description
			complete and all steps and DD statements have been processed.
PROC STATEMENT	P	Process_PROC	Occurs after a PROC statement is processed. It may be used to verify the name specified in the EXEC PROC statement. At this point, JOB/SCAN has already validated the EXEC PROC and the PROC statement and has stored all symbolics. It now validates the steps within the PROC.
SET STATEMENT	S	Process_SET	Occurs after a SET statement is processed and may be used to monitor and validate values assigned to symbolic parameters.
STEP END	Y	Process_End_Step	Occurs after step processing is complete and all DD statements for the current step are processed. It may be used to search for required DD statements.
WRAP UP	Z	Process_Final	The companion to Initialization always called when JOB/SCAN terminates. Use this pass to perform any necessary housekeeping functions.
XMIT STATEMENT	T	Process_XMIT	Occurs after an XMIT statement is processed. Remaining statements are then processed in the usual sequence.

## Using Data Elements from the EPL

The SJSSTDL library member EPLDOC describes the Extended Parameter List fields (EPL) available to REXX standards. To use the field names in a REXX exec, modify the name to meet REXX conventions: use the exact spelling and replace the separating hyphens (-) with underscores (\_). You can browse EPLDOC or print it using the JCL contained in SJSSTDL library member EPLPRNT. Refer to [EPL Field Examples](#) for additional information and examples. This table lists the COBOL names and REXX counterparts:

COBOL Field Name	REXX Name
EPLC-CALL-TYPE	EPLC_CALL_TYPE
EPLJ-JOBNAME	EPLJ_JOBNAME
EPLD-DDNAME	EPLD_DDNAME

The EPL data is accessed within the REXX exec by issuing external functions that are available to the REXX standards program. Use JSGET to fetch an EPL value into the REXX standards program. Use JSPUT to update an EPL value.



Data names in the Extended Parameter List that are associated with JOB/SCAN tables are not available using the JSGET or JSPUT functions. Refer to [Table Processing Overview and Procedure](#) for information on accessing JOB/SCAN tables.

### Example 1

Retrieve the EPL field EPLJ\_JOBNAME into the REXX variable named jobname. Code this statement within the "Process\_JOB" internal subroutine of the RKIT standards exec:

```
jobname = JSGET("EPLJ_JOBNAME")
```

### Example 2

Retrieve an 80-byte image of a control card. Code this statement within the "Process\_Control\_card" internal subroutine of the RKIT standards exec:

```
control_card = JSGET("EPL_CONTROL_CARD")
```

### Example 3

Update the EPL fields EPLC\_REQUEST\_JOB, and EPLC\_REQUEST\_DD\_ENTRY to indicate that job statements and DD statements are not to be processed by the REXX standards program. Code these statements within the Process\_Initial internal subroutine of the RKIT standards exec:

```
eplc_request_job = "N" /*Do not pass JOB statements to the exec*/
eplc_request_dd_entry = "N" /*Do not pass DD statements to the exec*/
call JSPUT "EPLC_REQUEST_JOB", "EPLC_REQUEST_DD_ENTRY"
```

## EPL Field Examples

EPL variables (defined in COBOL with OCCURS) are defined to REXX as stem variables.

### Example 1

COBOL:

```
10 EPLD-DSN-NODES.
15 EPLD-DSN-NODE OCCURS 20 TIMES PIC X(8).
```

REXX:

```
EPLD_DSN_NODE.1 through 20 are defined
EPLD_DSN_NODE.0 is not defined
```

### Example 2

COBOL:

```
10 EPLE-LIST-OF-ERRORS OCCURS 50 TIMES
INDEXED BY EPLE-SUB.
15 EPLE-MESSAGE PIC X(52).
```

```

      15 FILLER                PIC  X(07) .
15 EPLE-ERRTEXT-LVL PIC  X(01) .
15 EPLE-SEVERITY          PIC  S9(04) COMP.

```

REXX:

```

EPLMESSAGE.1      through 50 are defined
EPLERRTEXT LVL.1  through 50 are defined
EPLSEVERITY.1     through 50 are defined

```

## Table Processing Overview and Procedure

Table handling within the REXX Standards facility provides the following:

- Access to user-defined JSOPT Control Tables
- Capability to analyze, suppress, or change the severity of JOB/SCAN errors
- Capability to evaluate control card input

The REXX standards facility provides the JSTABLE function to access tables defined in the Extended Parameter List. The JSTABLE function is used to build a set of stemmed REXX variables. These stemmed variables can then be processed by the REXX standards program. For additional information on the JSTABLE function refer to [JSTABLE](#).

User-defined Control Tables (Ux) hold lists of site-specific values that are available to the REXX standards program. Because the actual table information is stored in the Control Table section of a PARMLIB member, the values are completely controlled by the user and are easy to modify. An example of how to access a user-defined table is included in the Process\_JOB internal subroutine of member RSMP in the SJJSTDL dataset. Refer to [Control Table Ux: Standards Facility User Table](#) and [User Control Table Processing](#) for information on user defined control tables.

JOB/SCAN Error Messages for each JCL statement are available to the REXX standards program. The JSTABLE function can be used to build a set of REXX stemmed variables that contain these error messages. The JSMSG function can be used to control the messages that appear on the Structured JCL Listing. Refer to the Process\_DD internal subroutine, in SJJSTDL library member RSMP, for an example of suppressing a JOB/SCAN generated message. Refer to [JSGET](#) and [JSTABLE](#) for information on using the JSMSG and JSTABLE external functions.

Control Cards accessed by JOB/SCAN are available to the REXX standards program. The control cards are stored in a table when JOB/SCAN calls the standards exec with the DD call type (for the control card file). A complete set of stemmed variables can be created from this table. Use the JSTABLE external function specifying CC to retrieve this information.

# REXX Programming Notes

This topic describes major REXX language components.

- Capitalization
- Input Parameters
- CALL\_TYPE Processing Subroutines
- User Control Table Processing
- Error Message Table Processing
- Control Card Table Processing
- Individual Control Over Messages
- Examining a Dataset Name
- Alternate Method for Examining a Dataset Name
- Global Variables
- Host Command Environment
- Compiling REXX Standards Programs
- Performance Considerations

## Capitalization

REXX translates all alphabetic characters to uppercase before evaluating an expression. Thus, you can use any mixture of upper and lower case letters in variable, function, and subroutine names.

Variable name arguments to JSGET, JSPUT, and JSTABLE are likewise translated to uppercase. It is not necessary to specify function arguments in uppercase.

## Input Parameters

REXX standards can accept run-time input parameters just like a standards program written in COBOL.

Suppose the user passes the input parameter string PARM1,PARM2 to REXX standards. The exec can retrieve the value of the parameter string with this REXX statement:

```
parm_string = ARG(1)
```

The variable parm\_string now contains the character string "PARM1,PARM2".

## CALL\_TYPE Processing Subroutines

When JOB/SCAN invokes a standards program, it sets the field EPLC\_CALL\_TYPE in the EPL with a 1-character code for the type of statement being processed. The statement might be a JCL statement, control card, JES statement, and so on.

All REXX standards should use JSGET to fetch the value of EPLC\_CALL\_TYPE from the EPL and branch to a subroutine for each type of call. See the SJJSTDL library member RSMP for an example of this technique (The below figure shows a portion of this member). Member EPLDOC in the SJJSTDL library contains complete descriptions of the various call types.

#### Sample of Call Type Processing in REXX Standards from RKIT and RSMP

```

/* Get call type. */
call_type = jsget("EPLC_CALL_TYPE")

/* Branch to appropriate processing routine. */
select
  when call_type = "I" then call Process_Initial
  when call_type = "J" then call Process_JOB
  when call_type = "S" then call Process_SET
  when call_type = "B" then call Process_JCLLIB
  when call_type = "2" then call Process_JES2
  when call_type = "3" then call Process_JES3
  when call_type = "C" then call Process_Comment_Card
  when call_type = "G" then call Process_Comment_Group
  when call_type = "L" then call Process_Control_Card
  when call_type = "N" then call Process_INCLUDE
  when call_type = "U" then call Process_End_INCLUDE
  when call_type = "R" then call Process_EXEC_PROC
  when call_type = "Q" then call Process_End_PROC
  ...
  when call_type = "K" then call Process_End_Job
  when call_type = "Z" then call Process_Final
  otherwise
    say "unknown call_type="call_type
end
...

```

## User Control Table Processing

User control tables are defined in the SJJSPARM dataset (usually in member JSOPT02) and may be accessed by a standards program. The tables are given names in the form Ux. The x in the Ux control table name identifies the specific table. Because x can take the value of any alphanumeric or national symbol, there can be a maximum of 39 Ux tables. The REXX function JSTABLE can be used to access any of the Ux tables.



JOB/SCAN stores the Ux tables rows in alphabetical order. This may differ from the order the SJJSPARM members use to define the tables.

### Example 1

This example shows how to query the existence of a UC table.

```

if JSTABLE("UC") > 0 then
  say "Row count in table UC="uc.0
else
  say "Table UC does not exist"

```

### Example 2

This code segment displays all the rows in the UA table.

```

call JSTABLE "UA"

```

```

if ua.0 > 0 then
  do i = 1 to ua.0
    say "UA row "i"="ua.i
  end
end

```

See [Control Table Ux: Standards Facility User Table](#) for complete details about user tables.

## Error Message Table Processing

The Error Message table contains the error messages generated by JOB/SCAN for the current JCL statement. If the current JCL statement does not contain any errors, the Error Message table is empty.

### Example 1

Count the total number of error messages generated by JOB/SCAN.

```

call_type = JSGET("EPLC_CALL_TYPE")
if call_type = "I" then /* INITIAL call? */
  do
    message_count = 0
    call JSPUT "message_count"
  end

  if JSTABLE("em") > 0 then
    do
      message_count = JSGET("message_count")
      message_count = message_count + em.0
      call JSPUT "message_count"
    end

  if call_type = "Z" then /* FINAL call? */
    do
      message_count = JSGET("message_count")
      say "Total number of messages generated="message_count
    end
  end
end

```

### Example 2

Change the severity of all JOB/SCAN-generated error messages to 8 (error).

```

if JSTABLE("em") > 0 then
  do i = 1 to em.0
    parse var em.i msg_number msg_flag msg_severity
    msg_severity = 8
    call JSMSG "change"
  end
end

```

## Control Card Table Processing

The Control Card table contains a copy of each card image read from a control card dataset. It is important to keep the following points in mind when processing the CC table:

- As JOB/SCAN processes a control card dataset, one row is added to the CC table for each control card.
- You can examine individual control cards when EPLC\_CALL\_TYPE is L.
- You can examine all control cards associated with a control card dataset when EPLC\_CALL\_TYPE is D (DD statement call).

### Example 1

Dump the contents of a control card dataset.



```

call_type = JSGET("EPLC_CALL_TYPE")
if call_type = "D" then /* DD statement? */
  if JSTABLE("CC") > 0 then
    do i = 1 to cc.0
      say "Control card " i " = " cc.i
    end
  end
end

```

## Example 2

Count the number of control cards processed by JOB/SCAN. Display the control card count in message DSS9076 in the Structured JCL Listing.

```

call_type = JSGET("EPLC_CALL_TYPE")
if call_type = "I" then /* INITIAL call? */
  do
    card_count = 0
    call_JSPUT "card_count"
  end
  if call_type = "D" then /* DD statement? */
    if JSTABLE("CC") > 0 then
      do
        card_count = JSGET("card_count")
        card_count = card_count + cc.0
        call_JSPUT "card_count"
      end
    end
  end
  if call_type = "Z" then /* FINAL call? */
    do
      card_count = JSGET("card_count")
      msg_level = " "
      msg_severity = 0
      msg_text = "076 Total control card count="card_count
      call JSMSG "ADD"
    end
  end
end

```

## Individual Control Over Messages

If the Standards program is to have individual control over multiple messages issued for the same DD statement, then the EPLC\_REQUEST\_INDIV\_ERMSG\_CNTL option must be set as follows in the REXX program:

```
EPLC_REQUEST_INDIV_ERMSG_CNTL = "Y"
```



The default for EPLC\_REQUEST\_INDIV\_ERMSG\_CNTL option is (N). With the default in force, suppose, for example, JOB/SCAN assigns multiple DSS6150 messages to one DD statement. If you mark this message for suppression, JOB/SCAN suppresses all the DSS6150 messages, and any change in severity to one affects the rest for the DD statement. However, with individual control asserted, JOB/SCAN suppresses or changes the severity of the targeted message occurrence only.

## Example

Instruct JOB/SCAN not to call your REXX exec for DD statements.

```

call_type = JSGET("EPLC_CALL_TYPE")
if call_type = "I" then /* INITIAL call? */
  do
    eplc_request_dd_entry = "N"
    call_JSPUT "eplc_request_dd_entry"
  end
end

```

## Examining a Dataset Name

It is sometimes necessary to examine the individual nodes within a dataset name. This section describes how to parse the full dataset name into individual nodes. See [Alternate Method for Examining a Dataset Name](#) for an alternative procedure.

A REXX standards program can retrieve the full 44-character dataset name with the following code:

```
call_type = JSGET("EPLC_CALL_TYPE")
if call_type = "D" then /* DD statement? */
do
  data_set_name = JSGET("EPLD_DSNAME")
end
```

To parse the dataset name into three nodes, use the REXX *parse* statement:

```
parse var data_set_name node1 "." node2 "." node3
```

This statement instructs REXX to parse the character string `data_set_name`. The first period or dot terminates the initial substring assigned to `node1`. Likewise, the second period or dot marks the end of `node2`. The last node, `node3`, contains the remaining characters from the dataset name.

This table illustrates the parsing results for various dataset names.

data_set_name	node1	node2	node3
"A"	"A"	""	""
"A.B"	"A"	"B"	""
"A.B.C"	"A"	"B"	"C"
"A.B.C.D"	"A"	"B"	"C.D"
"A.B.C.D.E"	"A"	"B"	"C.D.E"

Expanding the PARSE statement with more "." `noden` patterns places more individual nodes in separate variables.

```
parse var data_set_name node1 "." node2 "." node3 "." node4
```

This table show how the parse statement would handle A.B.C.D.E

data_set_name	node1	node2	node3	node4
"A.B.C.D.E"	"A"	"B"	"C"	"D.E"

## Alternate Method for Examining a Dataset Name

The previous section described how to parse the full dataset name into individual nodes. Alternatively, the EPLD\_DSN\_NODE stem variable can be used to retrieve each node.

A REXX standards program can retrieve the full 44-character dataset name with the following code:

```
call_type = JSGET("EPLC_CALL_TYPE")
if call_type = "D" then /* DD statement? */
  do i = 1 to 20 /* 20 possible nodes */
    varget_cmd = "dsnode.i = jsget("EPLD_DSN_NODE"."i")"
    interpret varget_cmd
  end
```

This table illustrates the parsing results for various dataset name:

dataset_name	node.1	node.2	node.3	node.4	node.5	...	node.n
"A"	"A"	""	""	""	""	""	""
"A.B"	"A"	"B"	""	""	""	""	""
"A.B.C"	"A"	"B"	"C"	""	""	""	""
"A.B.C.D"	"A"	"B"	"C"	"D"	""	""	""
"A.B.C.D.E"	"A"	"B"	"C"	"D"	"E"	""	""

## Global Variables

All REXX execs begin execution with empty variable pools and REXX standards are no exception. As new variables names are used, they are added to the local variable pool. Upon termination, the variable pool is destroyed.

JOB/SCAN calls REXX standards once per JCL statement. Each time, a new REXX execution environment is initialized. This means that the variable pool is re-initialized. Although this is the standard mode of operation for REXX, there might be times when you might want some variables to persist from one calling to another of the REXX standards. Consider these scenarios:

- Create a counter to record how many times JOB/SCAN uses the REXX standards. This counter needs to persist across all invocations of the program.
- Verify that each job step has a SYSUDUMP DD statement. A true/false flag would be initialized to false when an EXEC PGM statement is scanned. If a subsequent SYSUDUMP DD statement is found, the flag would be set to true. At end of step, if the flag is still false, then generate the appropriate error/warning message. In this scenario, the SYSUDUMP flag needs to persist for the duration of each job step.

In both examples, REXX the requirement is to keep track of information across multiple calls to REXX standards. One way to accomplish this is to use a file to store the required data values. While this method works for small files, it becomes I/O intensive for large JCL members and for large standards too; the overhead associated with file I/O makes this solution unattractive.

The alternative is to keep the values in memory outside the variable pool and in utilizing this mechanism, the JOB/SCAN REXX interface provides an effective solution. The JOB/SCAN REXX interface allows you to create a set of REXX variables that persist across invocations of the REXX standards program. These are called *global variables*. REXX functions are available for reading (JSGET) and writing (JSPUT) global variables.

### To use JSPUT and JSGET

1. Create a REXX variable.
2. Assign it a value.
3. Save the variable in the global variable pool with JSPUT.

REXX standards complete the processing for one statement and exits. JOB/SCAN calls the REXX standards again for the next statement.

4. Retrieve the saved variable from the global variable pool with JSGET; use and update the variable as appropriate.
5. Save the variable in the global variable pool with JSPUT for continued use.

### Creating Counters Example

In your standards program you may want to create various counters. This section describes how to initialize, increment, and reset counters. The technique illustrated using counters may be adopted to solve other programming problems.

Count the total number of DD statements processed.

- In the EPLC\_CALL\_TYPE = "I" subroutine, insert this code to initialize the counter:

```
/* Setup for INITIAL call */
num_dd = 0
call JSPUT "num_dd"
```

- In the EPLC\_CALL\_TYPE = "D" subroutine, insert the code to increment the DD statement counter.

```
num_dd = JSGET("num_dd")
num_dd = num_dd + 1
call JSPUT "num_dd"
```

- In the EPLC\_CALL\_TYPE="Z" subroutine, insert the display code.

```
num_dd = JSGET("num_dd")
say "Total number of DD statements processed="num_dd
```

### Restrictions

- The name of a global variable cannot exceed 80 characters. (This restriction is from IBM, and is also imposed by ASG.)
- A global variable cannot have the same name as an EPL field. (A global variable is specific to REXX; an EPL field is specific to JOB/SCAN. Global variables and EPL fields cannot have the same names because they use the same space in tables, which JOB/SCAN builds. If the same name is used by the global variable and the EPL field, the REXX interpreter will be confused.)
- The value of a global variable cannot exceed 200 characters. (This restriction is from IBM, and is also imposed by ASG.)

- Global variables persist for the duration of the JOB/SCAN run. Once created, a global variable cannot be deleted. (This restriction is imposed by ASG.)
- Only subscripted EPL variables are defined using REXX stem variables. Use the JSDUMP function to list the EPL fields available to REXX. See [JSDUMP](#) for more about JSDUMP.

## Host Command Environment

A REXX standards program executes in an MVS host command environment. This is a more restrictive environment than the standard TSO host command environment. For a detailed description of the MVS Host Command Environment, see *TSO/E V2 REXX/MVS Reference*. Executing under the MVS host command environment has these implications and restrictions:

- Be sure to preallocate all datasets that are used by the REXX exec, before going to REXX.
- See [Host Command Environment](#) for more information regarding REXX standards program restrictions.
- JOB/SCAN processes up to 10 JOBLIBs.
- A REXX standards program functions the same in all three JOB/SCAN execution modes: Edit macro, ISPF panel, and Batch job.
- The REXX standards program cannot execute commands in any other host command environment. Specifically, this means that the exec does not have access to the TSO environment when executing online. As an example, consider the following REXX statement that executes the TSO TIME command:

```
address TSO "TIME"
```

In a TSO environment, this statement produces the following output:

```
10 - address TSO "TIME"
    >>> "TIME"
TIME-02:30:52 PM. CPU-00:00:07 SERVICE-202327
SESSION-00:09:48
```

Statement 10 evaluates to the character string "TIME". This string is then passed to the TSO host command environment for execution.

In the non-TSO environment that the REXX standards program executes in, you get the following output:

```
10 - address TSO "TIME"
    >>> "TIME"
+++ RC(-3) +++
```

The -3 return code indicates that the command "TIME" was not found in the host command environment.

- These TSO/E REXX commands are available in the MVS Host Command Environment:
  - DELSTACK
  - DROPBUF
  - EXECIO
  - MAKEBUF
  - NEWSTACK
  - QBUF
  - QELEM
  - QSTACK
  - SUBCOM
  - TE
  - TS

## Compiling REXX Standards Programs

Sites with a REXX compiler might have the ability to compile their REXX standards.

### To compile REXX standards

1. Use the REXX compile PROC that was shipped with your REXX compiler.
2. Specify the CEXEC compiler option.
3. Specify the input (REXX standards program) on the SYSIN DD statement.
4. Specify the output (REXX executable) on the SYSEXEC DD statement.

JOB/SCAN can call the compiled version of REXX standards programs. Besides specifying the standards program name, you must also allocate the following in JS6PROC:

- The library containing the standards program (REXX executable) must be allocated to a SYSEXEC DD statement.
- The REXX LOADLIB must be added to the STEPLIB DD concatenation.



Depending on REXX installation options, you might not be able to execute these programs.

## Performance Considerations

A standards program could be called hundreds of times during the processing of a single set of JCL. Each time the program is invoked, JOB/SCAN passes a call type flag that indicates the type of call (EPLC\_CALL\_TYPE). You can improve the performance of your REXX standards program by telling JOB/SCAN to only invoke the program for certain JCL statements.

For example, suppose your standards program only performs standards checking on the JOB card. Instead of invoking the standards at least once for each JCL statement, you can instruct JOB/SCAN to call your standards only when it processes a JOB card. This dramatically reduces the number of times the exec is invoked (in this case JOB cards, only once per job).

Use JSPUT to update certain EPL fields that instruct JOB/SCAN when to pass control to your exec. Listed in this table are the EPL fields you can set to restrict when your exec is invoked. By default all of these fields are initialized to "Y" (for Yes).

### To instruct JOB/SCAN not to pass your exec a certain type of JCL statement

1. Set the corresponding EPL field to "N".
2. Call JSPUT to update these field in the EPL.

- EPLC\_REQUEST\_COMMENT\_CARDS

- EPLC\_REQUEST\_JCLLIB\_STATEMENT

- EPLC\_REQUEST\_CONTROL\_CARD

- EPLC\_REQUEST\_JES2\_STATEMENT

- EPLC\_REQUEST\_DD\_ENTRY
  - EPLC\_REQUEST\_END\_INCLUDE
  - EPLC\_REQUEST\_END\_JOB
  - EPLC\_REQUEST\_END\_PROC
  - EPLC\_REQUEST\_END\_STEP
  - EPLC\_REQUEST\_ENDIF
  - EPLC\_REQUEST\_EXEC\_PROC
  - EPLC\_REQUEST\_IF\_THEN
  - EPLC\_REQUEST\_INCLUDE\_STATEMENT
  - EPLC\_REQUEST\_JES3\_STATEMENT
  - EPLC\_REQUEST\_JOB
  - EPLC\_REQUEST\_OPERATOR\_COMMAND
  - EPLC\_REQUEST\_OUTPUT\_ENTRY
  - EPLC\_REQUEST\_PROC
  - EPLC\_REQUEST\_PROGRAM
  - EPLC\_REQUEST\_SET\_STATEMENT
  - EPLC\_REQUEST\_XMIT\_STATEMENT
-

# REXX Standards Errors and Abends

This topic explains about the REXX error messages and abends.

- [DSS08nn Error Messages](#)
- [RC\(-3\) Return Codes](#)
- [User Abends](#)
- [IRX0040I Error Messages](#)

## DSS08nn Error Messages

This topic explains the JOB/SCAN message formats and lists the messages in order by message ID.

### DSS0802

JSGET failed; unknown field name passed as input. Verify spelling of field name.

#### *Explanation*

JSGET was called to retrieve the value of an unknown field. Verify the spelling of the field name passed as an input parameter to JSGET.



If you are using JSGET to retrieve the value of an EPL field, remember that the REXX version of the EPL field names have underscores replacing hyphens.

### DSS0810

JSTABLE failed; calling argument must be a 2-character table name: "CC", "EM", or "Ux".

#### *Explanation*

JSTABLE was invoked with an invalid calling argument. There can only be one argument passed to JSTABLE, and that argument must be the 2-character name of the table to be loaded.

### DSS0811

JSTABLE failed; required input parameter not specified on call.

#### *Explanation*

JSTABLE requires one input parameter: the name of the table to retrieve.

### DSS0813

JSMSG failed; maximum of 50 messages per JCL statement exceeded.

#### *Explanation*

You can issue a maximum of 50 messages per JCL statement with JSMSG. If this limit is exceeded, an error is generated. You can monitor the number of messages currently issued by examining the return value from JSMSG. JSMSG returns



the number of messages issued for the current JCL statement.

#### DSS0814

JSMSG failed; single calling argument missing or invalid.

#### Explanation

A single calling argument is passed to JSMSG that specifies the operation to be performed. The valid arguments are "ADD", "FIND", "SUPPRESS", and "CHANGE". Make sure that your call of JSMSG includes one of these operation codes.

#### DSS0815

JSMSG fetch of REXX variables failed. IRXEXCOM return code=XXXXXXXX.

#### Explanation

JSMSG requires that certain REXX variables be initialized for each call type (for example, "ADD", "CHANGE", and so on). This error is generated when JSMSG detects a failure retrieving the values of the required REXX variables. IRXEXCOM is the routine used to fetch REXX variable values.

This table lists likely IRXEXCOM return codes.

Return code	Description
1	Variable not found (you did not initialize the required REXX variables)
4	Value truncated (the value of one of the REXX variables is too long - check the maximum variable lengths for the JSMSG variables)

#### DSS0816

JSMSG failed; invalid character passed for message level indicator. Verify MSG\_LEVEL value.

#### Explanation

MSG\_LEVEL is a 1-character variable that controls message output. Verify that you are initializing MSG\_LEVEL to one of the valid values.

#### DSS0817

JSMSG failed; message severity (MSG\_SEVERITY) must be an integer in the range 0 through 99.

#### Explanation

Message severity (REXX variable MSG\_SEVERITY) is a 1- or 2-digit non-negative integer. Check your initialization of MSG\_SEVERITY before the failing JSMSG call.

## RC(-3) Return Codes

If REXX cannot execute a command it issues a return code of -3 for the statement.

Consider the following example:

- A REXX exec has the following statement:  

```
JSPUT ("myvar1")
```

- When the exec is executed, these error messages are displayed:

```
12 - JSPUT("myvar")
    +++ RC(-3)
```

### Explanation

The error message indicates that statement number 12 in the exec caused the error. If you turn on the REXX trace in the exec with the statement TRACE RESULTS, the system displays the reason for the error.

```
12 - JSPUT("myvar")
    >>> "001"
    +++ RC(-3)
```

The REXX language processor evaluated the expression `jspu("myvar")`. The result of the evaluation was the character string "001" (one variable was successfully updated by JSPUT). REXX then tried to find a command in the host environment named "001". Since no command in the REXX's host environment was named "001", REXX issued a return code of -3.

### Fix

The RC(-3) error is caused by not invoking JSPUT as a subroutine or function. Change the invocation of JSPUT to:

```
call JSPUT "myvar"
```

## User Abends

This table show REXX User abends.

Abend Code	REXX Function	Load Module	CSECT	Description
U4001	(none)	J06YREX2	D06YREX2	Caller did not supply an input parameter list. This module requires one input parameter; the address of the EPL.
U4002	(none)	J06YREX2	D06YREX2	Invalid EPL. EPL field "C2LITERL" did not contain the character string C'DSSI EPL END'.
U4003	JSGET	J06YREFP	D06YREGT	Address of "User Area" in REXX environment block is zero.
U4004	JSGET	J06YREFP	D06YREGT	Contents of "User Area" eye-catcher has changed.
U4005	JSPUT	J06YREFP	D06YREPT	Address of "User Area" in REXX environment block is zero.
U4006	JSPUT	J06YREFP	D06YREPT	Contents of "User Area" eye-catcher has changed.
U4007	JSMSG	J06YREFP	D06YREMG	Address of "User Area" in REXX environment block is zero.
U4008	JSMSG	J06YREFP	D06YREMG	Contents of "User Area" eye-catcher has changed.

Abend Code	REXX Function	Load Module	CSECT	Description
U4009	JSDUMP	J06YREFP	D06YREDP	Address of "User Area" in REXX environment block is zero.
U4010	JSDUMP	J06YREFP	D06YREDP	Contents of "User Area" eye-catcher has changed.
U4011	JSTABLE	J06YREFP	D06YRETB	Address of "User Area" in REXX environment block is zero.
U4012	JSTABLE	J06YREFP	D06YRETB	Contents of "User Area" eye-catcher has changed.
U4013	(none)	J06YREX3	D06YREX3	Caller did not supply an input parameter list. This module requires one input parameter; the address of the Exit Parameter List (EPL).
U4014	(none)	J06YREX3	D06YREX3	Invalid Exit Parameter List. EPL field "C2LITERL" did not contain the character string C'DSSI EPL END'.

## IRX0040I Error Messages

REXX issues error message IRX0040I when an invalid call is made to a function or subroutine.

When invalid parameters are passed to one of the JOB/SCAN REXX functions, the exec fails with error message IRX0040I.

### Example

A REXX standards program fails with the following error:

```
15 +++ y = JSGET("myvar1")
IRX0040I Error running RXSAMP10, line 15: Incorrect call to routine
```

The most likely cause of the error is the argument passed to JSGET. Check to make sure that `myvar1` is the name of a REXX variable that was previously saved with a JSPUT.

# REXX Functions

This topic describes the REXX functions used by the standards facility.

- JSGET
- JSPUT
- JSMSG
- JSTABLE
- JSDUMP

## JSGET

JSGET fetches the value of an EPL field or global variable.

There are two ways JSGET can be called:

- Fetch the value of a single EPL field or global variable, .
- Fetch all EPL fields and global variables, .



For performance reasons ASG recommends that individual values of fields and variables be retrieved as needed with JSGET.

### JSGET Syntax

JSGET Syntax 1:

```
variable_name = JSGET("field_name|"global_var")
```

The argument to JSGET, `field_name` or `variable_name`, must be the name of an EPL field or global variable, respectively. JSGET assigns the current value of `field_name` or `global_var` to `variable_name`. JSPUT must have previously set the REXX variable `global_var` in order for JSGET to fetch it.

JSGET Syntax 2:

```
CALL JSGET
```

A call to JSGET without any calling argument causes the REXX interface to create and initialize a REXX variable for *every* EPL field and *every* global variable. For example, after executing `CALL JSGET`, the REXX variable `EPLC_CALL_TYPE` contains the value of the corresponding EPL field.

### JSGET Examples

After this statement is executed:

```
var1 = JSGET("EPLC_CALL_TYPE")
```

the REXX variable `var1` contains the value of the EPL field named `EPLC_CALL_TYPE`

This statement creates and initializes every EPL variable and every global variable:

```
CALL JSGET
```

A REXX standards can then perform checks upon any of the EPL fields:

```
IF EPLD_DDNAME = "SYSTSPRT" THEN ...
```

## JSPUT

JSPUT is used to update an EPL field or to save a value to a global variable. Local variables saved with JSPUT become global variables; to change the global value, use JSPUT and the variable name; to change the local value, assign it one using normal REXX assignment.

### JSPUT Syntax

```
CALL JSPUT var1, var2, ...
```

At least one argument must be passed to JSPUT.

- If the argument is the name of an EPL field, the JSPUT updates the actual EPL field.
- If the argument is not the name of an EPL field, JSPUT assumes it is the name of a global variable. JSPUT saves the current value of the variable in the global variable pool.

### JSPUT Return Value

JSPUT returns a 3-digit count of the number of REXX variables successfully updated.

### JSPUT Examples

JSPUT Example 1:

This code segment shows how to use the global variable pool to keep a count of the number of times the REXX exec was called.

---

```
call_type = JSGET("EPLC_CALL_TYPE")
if call_type = "I" then /* INITIAL call? */
do
  call_count = 1
  call JSPUT "CALL_COUNT"
end
else
do
  call_count = JSGET("CALL_COUNT") /* Increment counter */
  call_count = call_count + 1
  call JSPUT "CALL_COUNT"
end
if call_type = "Z" then /* FINAL call? */
say "Total call count for REXX exec="call_count
```

---

JSPUT Example 2:

```
new_var = 1
call JSPUT "new_var"
new_var = 2
```

When JSPUT is called, it saves the current value of the REXX variable `new_var` in the global variable pool. If the REXX variable is later updated in the REXX exec, JOB/SCAN does not automatically update its value in the global variable pool.

In this example, even though `new_var` was set to 2, it still has the value of 1 in the global variable pool. JSPUT is the only way to save variables in the global variable pool.

## JSMSG

JSMSG is used to control the messages that appear in the Structured JCL Listing.

Use JSMSG to:

- Add a user-generated message to the Structured JCL Listing and, optionally, its extended message text.
- Determine if JOB/SCAN generated a specific message.
- Cancel a specific JOB/SCAN-generated message for a single JCL statement.
- Change the severity of a JOB/SCAN-generated message.

### JSMSG Syntax

```
CALL JSMSG "call_type"
```

The input argument `call_type` determines the processing JSMSG performs. Furthermore, each `call_type` requires values be pre-assigned to one or more special variables prior to the call to JSMSG. This table lists and briefly describes the required variables for each `call_type` and the contents of those variables; see next table for a full description of the REXX variables.

Call Type	Description	Required REXX Variables
ADD	Inserts a user-generated message into the SJL.	MSG_TEXT - contains the three-digit message number (e.g. 070) and the message text. MSG_LEVEL - control style of message output (short message or extended text message). MSG_SEVERITY - severity code to assign to message.
FIND	Looks for a JOB/SCAN-generated message.	MSG_NUMBER - number of message to search for. FIND will not search for user messages (i.e. DSS9nnn).
SUPPRESS	Cancels a JOB/SCAN-generated message.	MSG_NUMBER - number of message to suppress. SUPPRESS will not suppress user messages (i.e. DSS9nnn).
CHANGE	Alters the severity of a JOB/SCAN-generated message.	MSG_NUMBER - number of message of which to change severity code. CHANGE will not change user messages (i.e. DSS9nnn). MSG_SEVERITY - new message severity code.

This table describes the REXX variables.

REXX Variable	Description
MSG_TEXT	A 52-character string composed of a 3-digit message number in the range 000-999 followed by up to 49 characters of text. Pad one- and two-digit numbers with leading zeros.
MSG_LEVEL	<p>A 1-character message level indicator that controls the style of message output on the Structured JCL Listing.</p> <p>"S" Displays only the short message.</p> <p>"X" Displays both the short message and the extended text.</p> <p>" " Uses the value specified for the SJL Report Option to control the display of extended text.</p>
MSG_SEVERITY	<p>A number in the range 0 to 99 that denoting the severity of the message:</p> <ul style="list-style-type: none"> <li>0 Advisory (A)</li> <li>1-4 Warning (W)</li> <li>5-19 Error (E)</li> <li>20-99 Catastrophic (C) (causes JOB/SCAN to terminate processing)</li> </ul> <p>JOB/SCAN automatically appends A, W, E, or C to the message number in the SJL.</p>
MSG_NUMBER	<p>A 7-character JOB/SCAN message number.</p> <p>Format: DSSnnnn, where nnnn is an integer. Do not include the message severity appendix.</p>




See [JSMSG Examples](#) through for information on how JOB/SCAN uses MSG\_TEXT and MSG\_SEVERITY to generate the actual message in the Structured JCL Listing.

### JSMSG Return Value

This table lists the JSMSG return values by call type in the special variable RESULT.

Call Type	Return Value
"ADD"	JSMSG returns the current number of user-generated message queued for the Structured JCL Listing as a 3-digit integer. If it is possible that your REXX standards might issue a large number (50 is the maximum number per invocation) of messages per JCL statement, use this return value to monitor the JSMSG return value.

Call Type	Return Value
	<div style="border: 1px solid purple; padding: 5px;">  If a 51<sup>st</sup> message is issued, JOB/SCAN generates message DSS0813 and REXX standards fails with the REXX error IRX0040I. </div>
"FIND"	A 1 (true) is returned if the requested message number was found. Otherwise, a 0 (false) is returned.
"SUPPRESS"	A 1 (true) is returned if the requested message was successfully suppressed. Otherwise, a 0 (false) is returned.
"CHANGE"	A 1 (true) is returned if the severity of the requested message was successfully changed. Otherwise, a 0 (false) is returned.

## JSMMSG Examples

### JSMMSG Example 1

```
msg_text = "001 JOB CLASS IS NOT VALID FOR PRODUCTION"
msg_level = " "
msg_severity = 0
call JSMMSG "ADD"
```

This message is generated in the Structured JCL Listing:

```
*ADVISORY - DSS9001A - JOB CLASS IS NOT VALID FOR PRODUCTION
```

The first 3 characters of the message text are inserted into the message number DSS9nnn. A severity of 0 indicates an advisory message; see the \*ADVISORY indicator and the "A" suffix on the message number.

### JSMMSG Example 2

```
msg_text = "123DSNAME MUST START WITH APPLICATION CODE"
msg_level = " "
msg_severity = 4
call JSMMSG "ADD"
```

This message is generated on the Structured JCL Listing:

```
**WARNING - DSS9123W - DSNAME MUST START WITH APPLICATION CODE
```

The first 3 characters of the message text are inserted into the message number DSS9nnna. A severity of 4 indicates a warning; see the \*\*WARNING indicator and "W" suffix on the message number.

### JSMMSG Example 3

```
msg_text = "AB567 - CLASS PARAMETER REQUIRED"
msg_level = " "
msg_severity = 6
call JSMMSG "ADD"
```

This message is generated on the Structured JCL Listing:

```
***ERROR - DSS9999E - AB567 - CLASS PARAMETER REQUIRED
```



The default message number of DSS9999 E is generated because the first 3 characters of `msg_text` are not numeric. A severity of 6 indicates an error; see the `***ERROR` indicator and the "E" suffix on the message number.

#### JSMMSG Example 4

This example demonstrates how to monitor the number of messages issued by JSMMSG.

```
msg_text = "012 - JOB CLASS IS NOT VALID FOR PRODUCTION"
msg_level = " "
msg_severity = 4
call JSMMSG "ADD"
msg_count = RESULT
say "Current message count="msg_count
call JSMMSG "ADD"
msg_count = RESULT
say "Current message count="msg_count
```

The first `SAY` statement displays a count of "001". For the second `SAY` statement the count is reported as "002".



RESULT is the special REXX variable containing the value returned by a subroutine call.

#### JSMMSG Example 5

Change the severity of JOB/SCAN message DSS4380 to an 8 (error).

```
msg_number = "DSS4380"
msg_severity = 8
call JSMMSG "CHANGE"
```

#### JSMMSG Example 6

Suppress the display of JOB/SCAN error message DSS4050.

```
msg_number = "DSS4050"
call JSMMSG "SUPPRESS"
```

#### JSMMSG Example 7

Find whether JOB/SCAN message 1080 occurred and change the severity from 8 to 12. The calls to `FIND` and `CHANGE` use the same `msg_number`; so it occurs once for both call types.

```
msg_number = "DSS1080"
call JSMMSG "FIND"
if RESULT = 1 then
then
do
  msg_severity = 12
  call JSMMSG "CHANGE"
end
```

## JSTABLE

A call to `JSTABLE` loads a set of *stemmed* REXX variables with the contents of a JOB/SCAN table. `JSTABLE` can be called to load the JOB/SCAN tables listed in this table:

Table Type	Description
Ux	The Ux control tables are defined in the SJJSPARM dataset (usually in member JSOPT02). The x in

Table Type	Description
	the Ux control table name identifies the specific table. Since x can take the value of any alphanumeric or national symbol, there can be a maximum of 39 Ux tables.
CC	A Control Card table is created when a DD statement identifying a control card dataset is scanned by JOB/SCAN. The CC table contains a row for each statement in the corresponding control card dataset.
EM	The Error Message table contains the error messages generated by JOB/SCAN for the current JCL statement. If the current JCL statement does not contain any errors, the Error Message table is empty.



JSTABLE creates a local REXX copy of the requested table. Any changes made to the REXX copy of the table are not propagated back to the actual table.

### JSTABLE Syntax

```
CALL JSTABLE table_name
```

Where `table_name` is the 2-character name of the JOB/SCAN table to be loaded:

Type of Table Request	<code>table_name</code> value
Ux table	"UA", "UB", ..., "UZ" "U0", "U1", ..., "U9" "U@", "U#", "U\$"
Error Message table	"EM"
Control Card table	"CC"

After calling JSTABLE, JOB/SCAN creates the REXX stem-variables listed in this table:

REXX Variable	Description
<code>table_name.0</code>	Contains the number of rows in the requested table. If the requested table does not have any rows (or does not exist), then <code>table_name.0</code> is set to zero.
<code>table_name.1</code>	Contains the contents of the first row in <code>table_name</code> .
<code>table_name.2</code>	Contains the second row in <code>table_name</code> .

REXX Variable	Description
<code>table_name.i</code>	Contains the i-th row in <code>table_name</code> .

This table describes the characteristics of the returned table.

Table Type	Length of a row variable	Comments
Ux	65	The Ux row returned by JSTABLE corresponds to character positions 8-72 on the PARMLIB statement defining the table. The Ux table is sorted in ascending order.
EM	12	An EM table row has the following format: <message number> <action flag> <severity code> Blanks separate the three fields. See <a href="#">through</a> for sample code that parses the returned EM table row.
CC	80	The full 80-character card image is returned for the Control Card table. The rows in the Control Card table match the order of the statements in the control card dataset.

### JSTABLE Return Value

JSTABLE returns the number of rows in the requested table. The return value can be examined in two ways:

- If JSTABLE is invoked as a subroutine (for example, `CALL JSTABLE "CC"`), the return value is saved in the REXX special variable `RESULT`.

Or

- If JSTABLE is invoked as a function (for example, `X = JSTABLE("CC")`), the function return value can be examined directly and immediately.

### JSTABLE Examples

#### JSTABLE Example 1

This example displays all cards contained in a control card dataset.

```
if jstable("CC") > 0 then
  do i = 1 to cc.0
    say "CC row "i"="cc.i
  end
```

#### JSTABLE Example 2

For a given 4-character application code, this example examines the UA application table to see if the code is valid.

```
code to test = "TEST"
valid code = "NO"
call JSTABLE "UA"
```

```

do i = 1 to ua.0
  if code_to_test = substr(ua.i,1,4) then
    do
      valid_code = "YES"
      leave
    end
  end
end
if valid_code = "YES" then
  say "Found got a valid application code"
else
  say "application code is not valid"

```

---

### JSTABLE Example 3

This example shows two ways to determine if a JOB/SCAN error message has been issued for a JCL statement.

```

/* Technique 1: Issue a SEARCH against the Error Message table */
msg number = "DSS4050"
if JSMMSG("FIND") then
  say "JOB/SCAN error message DSS4050 found"
else
  say "JOB/SCAN error message DSS4050 not found"
/* Technique 2: Manually search the Error Message table */
call JSTABLE "EM"
message_found = "NO"
do i = 1 to em.0
  parse var em.i mnumber mflag mseverity
  if mnumber = "DSS4050" then
    do
      message_found = "YES"
      leave
    end
  end
end
if message_found = "YES" then
  say "JOB/SCAN error message DSS4050 found"
else
  say "JOB/SCAN error message DSS4050 not found"

```

---

## JSDUMP

The JSDUMP function dumps the names and values of all EPL fields and global variables.

JSDUMP output is written to the same file as SAY instruction output.

### JSDUMP Syntax

```
CALL JSDUMP
```



Use this function with caution, because it dumps the entire variable pool stepping through every stem variable.

# SJSSTDL Library Members

This topic provides information on how to use SJSSTDL library members and explains about the SJSSTDL library members related to implementing site standards.

## Using SJSSTDL Library Members

The standards programs should be compiled and linked to the JOB/SCAN load library or a user load library that is concatenated with the JOB/SCAN load library (ddname JLLD). JOB/SCAN loads and calls standards programs at run time. Do not link standards programs with the JOB/SCAN load module.

## What the SJSSTDL Library Contains

This table explains the SJSSTDL library members related to implementing site standards. Some of these members are sample skeleton programs upon which site-specific requirements can be based. Standards programs can be written in any language including Enterprise COBOL, REXX, and Assembler. Sample programs are available in the SJSSTDL library.

Member	Contents
AEPL	Assembler mapping macro defining data elements that pass information between the validation routine and the standards program. As JOB/SCAN runs, it places appropriate values from the JCL into these field names. Do not alter any data elements defined in this member.
ASMP	Sample Assembler program that performs several typical site standard validations.
CCHK	COBOL Copy member that validates whether the correct version of CEPL is in use. This copy member should be included in the initial call section of the program. If the wrong version of CEPL is invoked, CCHK returns error messages with a severity level of 20 to JOB/SCAN; processing discontinues.
CEECMPLK	Compile and link ASG-JOB/SCAN standards programs using enterprise COBOL compiler for z/OS.
CEPL	COBOL Extended Parameter List defining data elements that pass information between the validation routine and the standards program. As JOB/SCAN runs, it places appropriate values from the JCL into these field names. Do not alter any data elements defined in this member.
CERR	COBOL copy member containing a procedure for transferring error messages from the standards program to JOB/SCAN. JOB/SCAN permits up to 50 errors per statement call, and automatically includes these messages in the Structured JCL Listing. (See for information on processing more than 50 errors.) WE DON'T DISCUSS HOW TO PROCESS > 50 ERRORS
CSMP	Sample Enterprise COBOL program that performs several typical site standard validations. If a user-written standards program does not work as anticipated, run JOB/SCAN specifying this program as the standards program. If CSMP works, there may be a logic error in the standards program. If CSMP does not work, contact the Product Administrator.

Member	Contents
CTRKIT	Starter kit for Standards Prompter; also used to perform exercises in training classes. Because the Prompter uses this member to build COBOL standards, be sure to make a copy of CTRKIT if you need to make changes to it.
C2SMP	Sample Enterprise COBOL program that performs several typical site standard validations. If a user-written standards program does not work as anticipated, run JOB/SCAN and specify this program as the standards program. If C2SMP works, there may be a logic error in the standards program. If C2SMP does not work, contact the Product Administrator.
EPLDOC	Member containing detailed definitions of all fields in the CEPL. Designed for online browsing; provided to assist in writing of standards programs.
EPLPRNT	Member containing JCL to print a hardcopy of ELPDOC.
JSVERIFY	Member containing JCL to run Standards Facility field display program to assist with debugging a standards program.
JSXTDG0	COBOL standards program that checks GDG-ALL datasets for generations, which may have been migrated. Calls assembler subroutine JSXTDG1.
JSXTDG1	Assembler subroutine that performs actual validation of GDG-ALL datasets for generations that may have migrated. Called by COBOL program JSXTDG0; it must run in the TSO environment because it invokes the TSO Service facility.
JSXTDJES	COBOL standards program used to generate the JSOPTJ2 member in the SJJSPARM library. This program reads PROCLIB DD statements found in PROC00, and generates appropriate File Allocation statements in JSOPTJ2. See <a href="#">Reference Information</a> .
RKIT	Sample REXX program shell for Standard Facility. This is an example of how site-specific logic can be added to the RKIT shell.
RSMP	Sample REXX program used with the JOB/SCAN Standards Facility; contains examples.

# Call Types

The Standards Facility processes call types as described in this table. Code designed to perform site-specific validation of the JCL statement parameters should be added to the appropriate call type sections described here. Each call type is described and paired with the applicable standards program section title.

Call Type	Standards Program Section	Description
Initialization	EPLC-CALL-INITIAL	Always called when JOB/SCAN is initialized and before any JCL is validated. Use this level to set values for subsequent calls and to initialize any special switches used by these subsequent calls.
Wrapup	EPLC-CALL-FINAL	Companion to Initialization; always called when JOB/SCAN terminates. Use this level to perform any necessary housekeeping functions.
Job start	EPLC-CALL-JOB	Occurs after the JOB statement is processed; may be used to validate JOB statement values, job naming conventions, etc.
Job end	EPLC-CALL-END-JOB	Occurs after all JCL statements for the current job are processed. Any necessary switches required by the next job can be reset in this section.
SET statement	EPLC-CALL-SET-STATEMENT	Occurs after a SET statement is processed; may be used to monitor and validate values assigned to symbolic parameters.
JCLLIB statement	EPLC-CALL-JCLLIB-STATEMENT	Occurs after a JCLLIB statement is processed. Up to 10 library names are passed for validation and monitoring.
JES2 statement	EPLC-CALL-JES2-STATEMENT	Occurs after a JES2 statement is processed; may be used to validate parameters and options specified on the JES2 statement.
JES3 statement	EPLC-CALL-JES3-STATEMENT	Occurs after a JES3 statement is processed; may be used to validate parameters and options specified on the JES3 statement.
Comment statements	EPLC-CALL-COMMENT-CARDS	Occurs after a <code>/**</code> comment statement is processed; may be used to verify the contents of <code>/**</code> comment statements.
INCLUDE start	EPLC-CALL-INCLUDE-STATEMENT	Occurs after an INCLUDE statement is processed; may be used to verify the member name specified in the INCLUDE statement.
INCLUDE end	EPLC-CALL-END-INCLUDE	Occurs after JOB/SCAN has processed all statements for the

Call Type	Standards Program Section	Description
		current INCLUDE.
PROC statement	EPLC-CALL-PROC	Occurs after a PROC statement is processed; may be used to verify the name specified in the EXEC PROC statement. At this point, JOB/SCAN has already validated the EXEC PROC and the PROC statement, and has stored all symbolics; now validates steps within the PROC.
PROC end	EPLC-CALL-END-PROC	Occurs after all JCL statements for the current PROC are processed; may be used to verify programs executed by the PROC. At this point, validation of the PROC is complete, and all steps and DD statements have been processed.
IF-THEN start	EPLC-CALL-IF-THEN	Occurs after an IF statement is processed. At this point, IF processing is not complete.
ELSE statement	EPLC-CALL-ELSE	Occurs after an ELSE statement is processed. At this point, validation of the IF statement is not complete.
ENDIF statement	EPLC-CALL-ENDIF	Occurs after an IF-THEN-ELSE statement group is processed. At this point, validation of the IF-THEN-ELSE statement group is complete.
XMIT statement	EPLC-CALL-XMIT-STATEMENT	Occurs after an XMIT statement is processed; remaining statements are then processed in the usual sequence.
EXEC PGM statement	EPLC-CALL-PROGRAM	Occurs after an EXEC PGM= statement is processed; may be used to verify step names, program names, region parameters, etc.
EXEC PROC statement	EPLC-CALL-EXEC-PROC	Occurs after an EXEC PROC= statement is processed; may be used to validate any parameters specified on the EXEC PROC= statement. At this point, JOB/SCAN has stored only the symbolics specified on the EXEC PROC= statement; symbolics specified on the PROC PROTO-TYPE statement are available at EPLC-CALL-PROC time.
STEP end	EPLC-CALL-END-STEP	Occurs after step processing is complete and all DD statements for the current step are processed; may be used to search for required DD statements.
DD statement	EPLC-CALL-DD-ENTRY	Occurs after each DD statement is processed; may be used to validate naming conventions for ddnames, dataset names, and



Call Type	Standards Program Section	Description
		unit names. Also checks VOLSERS, DCB information, SPACE allocations, SYSOUT parameters, DISP, DSORG, referbacks, etc.
OUTPUT statement	EPLC-CALL-OUTPUT-ENTRY	Occurs after an OUTPUT JCL statement is processed; may be used to verify standards on this statement.
Control cards	EPLC-CALL-CONTROL-CARD	Occurs for each control card; may be used to validate the contents of the control cards.
Control card end-of-file	EPLC-CALL-END-CONTROL-CARDS	Occurs after all control cards have been read. May be used to get the total number of control cards associated with a DD statement.
User Errors for any of these call types	EPLC-CALL-BACK-USER-ERROR	Standards Facility is automatically called back with this call type when 50 user errors are encountered. Checking this field allows additional errors to be issued.

## Common Questions about Writing Standards

This topic answers questions that are frequently asked about writing standards for JOB/SCAN.

### ***May I use the Prompter on a standards program that I have edited with ISPF edit?***

You may but with restrictions. The Prompter creates a COBOL program in a fixed format. This allows the Prompter to display the individual standards for each section. If you stay within the format of the Prompter-generated member and change *none* of its comment lines, you may add, delete or change code through the Prompter. Also, the Prompter does not display more than 7 lines of COBOL no matter how many are coded.

### ***How do I know what field names to use when comparing the various parameters that I need to validate?***

Most fields that contain values from parameters have a name similar to the parameter in question. These fields are defined in the SJSSTDL library member CEPL. For example:

- The value of the NOTIFY parameter from the JOB statement can be found in a field called EPLJ-NOTIFY.
- The value of the DSN parameter on a DD statement can be found in a field called EPLD-DSN.
- The value of the PGM parameter on an EXEC statement can be found in a field called EPLX-PGM.

You can see from these examples that the last letter of the prefix in these fields indicate which statement type the field belongs to. The last letter of the suffix "J", "D", and "X" refers to the JOB, DD, and EXEC statements, respectively.

Every possible statement type and all parameters coded in the JCL are made available to the standards module.

### ***Where can I get a full listing of the parameter fields that are available?***

The complete list of parameters that are available by statement type and their lengths can be found in member CEPL in the SJSSTDL library, which is distributed with JOB/SCAN.

### ***How do I find out what I can expect to see in each field, their lengths, and the defaults if the parameter was not coded in the JCL?***

The member EPLDOC, also in the SJSSTDL library, contains complete documentation on each field in the parameter list. There is also another member, EPLPRINT, in the SJSSTDL library that can be used to generate a hard copy listing of this documentation, if desired.

### ***What other tools do I have available to me that can help?***

The JSVERIFY member in the SJSSTDL library provides a handy utility that displays the contents of each field in the test JCL that is being processed by JOB/SCAN. This can be used to resolve any questions that you might have regarding the value of any field that is passed to your standards module.

### ***From time to time, I might need to verify some parameter content (such as the first node of a DSN) against a list of valid application IDs. Do I have to code a bunch of IF-THEN statements in my standards module? What happens if some of them change?***

JOB/SCAN provides a number of functions that support this type of need. The easiest is to place the list of valid

application IDs in a special control file that is part of JOB/SCAN. After this is done, you can place statements in your standards module that compare the JCL item in question, in this case the first node of the DSN, to the list stored in JOB/SCAN, and generate an error message if a match is not found. A sample for this lookup process is provided in the SJSSTDL library.

There might be other cases where you might already have a file in your system that contains the valid application IDs. In this situation, you might decide that you would rather read that file to validate application IDs. This is also supported by JOB/SCAN because its architecture uses standard IBM structures and file handling functions. You can define an input file using the example in [ENVIRONMENT DIVISION](#) which demonstrates defining an output file.

***When I find a standard that has been violated, I would like to be able to change the JCL that is not in conformance with my standards as opposed to generating an error message. How do I do that?***

JOB/SCAN can be used to bring JCL that violates your standards into conformance. There are two types of corrections that can be made using two facilities of JOB/SCAN that were designed for this purpose.

The first situation is when there is a given set of constraints that are applicable for all production jobs. For example, the JOB name of a production job must begin with the letter 'P', it must run in CLASS=P; NOTIFY=userid is not allowed on production jobs. All datasets that are being created must be on SMS control, BLKSIZE, if coded must be set to 0, etc.

The easiest way to make JCL conform to these types of standards is to use the JCL Change facility of JOB/SCAN. Because the standards listed above do not require extensive analysis, they are a perfect candidate for automatic correction.

There are other types of violations, however, that might require extensive analysis, perhaps consulting with another person, and even an educated guess. Some examples of these might be BLKSIZE not a multiple of the LRECL, first node of the DSN not matching any of the valid application IDs, duplicate DDname, etc. The problem with trying to correct these is obvious. In the first case, is the LRECL or the BLKSIZE correct? In the second case, what should the first node be? The application ID that it most closely resembles? In the third case, which one of the duplicate DDs should be deleted?

Because of these considerations, you need to be careful about which problems you want to correct automatically. When you decide that there are some that you feel confident that can be automated, the Standards Facility and the JCL Change function can be combined to perform any kind of automatic corrections that are appropriate.

***We have different standards for different purposes. For example, each division in our company requires unique standards that apply only to their processing environment. We might even have different applications that require different standards. Can JOB/SCAN support these varying needs easily? Do we have to have different versions of JOB/SCAN installed?***

JOB/SCAN was designed with consideration for differing needs such as this one. We have a saying that goes "different standards for different folks!" What that means is that you can have any number of standards modules that can be activated by the user at execution time. Merely specify which standards module is applicable to this run

and JOB/SCAN does the rest.

***In our environment, we don't like to give end users the ability to run JOB/SCAN without the standards being invoked. Can I define my standards and lock everyone into using them all the time?***

JOB/SCAN provides a facility that lets you define exactly the way you want to implement standards. For some users, you might wish to always force the use of a specific standards module. For others, more flexibility is desired. The bottom line is that your implementation can range from the strictest control to complete freedom of choice and any level in between.

***Our system programming group is very concerned about the impact software products such as JOB/SCAN can potentially have on the system due to usage throughout the company. Will JOB/SCAN slow down our system, especially when standards are added?***

JOB/SCAN has been designed overall with efficient execution in mind. Specifically, the Standards Facility has been implemented using not only the most flexible architecture but also the most efficient. When you run JOB/SCAN side by side with other products, you will see that JOB/SCAN performs significantly more efficiently.

***At our site, there are some JCL errors that are accepted by our MVS system because we have modified the system. Can JOB/SCAN be tailored to accept this type of JCL as valid and not generate an error message?***

Absolutely. JOB/SCAN provides several levels of error suppression. You might decide that a certain error is never applicable in your environment. In this case, you can tell JOB/SCAN not to generate an error message under any circumstances. There might be other situations where the criteria is more focused. For example, if PGM010 is executing and the DDname is //DD010 do not generated error number DSS4050. There might be yet other situations that require further analysis before suppression takes place. JOB/SCAN easily supports these as well through the use of the Standards Facility.

***At our site we use some "foreign" statements that don't look like JCL statements. We use pre-processors that handle those statements before the MVS reader interpreter sees the JCL. Can I use the Standards Facility to suppress any errors that might result from these?***

You could do that but it is not necessary. JOB/SCAN has a special facility that is designed specifically for this purpose. It is an "input filter" that can be used to either delete, modify, or comment out "foreign" statements before JOB/SCAN starts processing the JCL.

# Messages and Codes

This documentation provides information about the messages and return codes generated by ASG-JOB/SCAN (herein called JOB/SCAN) and ASG-DOCU/TEXT (herein called DOCU/TEXT).

JOB/SCAN is a JCL asset management tool that reduces syntactic and production errors. It does this by simulating the MVS run-time environment to validate the syntax of JCL and many other control statement types.

DOCU/TEXT is an optional add-on component to JOB/SCAN that produces reports that document systems and jobs, such as cross-reference reports that show “where-used” lists of jobs, PROCs, programs, and files.

Topic	Description
<a href="#">System Messages</a>	Explains the product message formats and lists the JOB/SCAN messages in alphabetical order by message ID.
<a href="#">Return Codes</a>	Lists and explains the product return codes.

---

# System Messages

This topic explains message formats and all product messages for JOB/SCAN and DOCU/TEXT in order by message ID. For explanations of product return codes, see [Return Codes](#).



## Message Format

This section contains a complete listing of all product error messages in order by message number.

Two types of error messages are listed:

- Product execution messages.
- Verification messages generated by the product as your JCL is processed.

Product messages have this general format:

Number	Meaning
DSS	Standard prefix for all product messages.
10000 <	<p>Product execution messages. These messages describe possible reasons for why the product did not run. These types of messages can appear in the started task (SYSPRINT DD), JPSEL, or SJL.</p> <div style="border: 1px solid purple; padding: 5px; margin: 5px 0;"> <p> Messages numbered less than 10000 cannot be viewed and the severity level of these messages cannot be modified.</p> </div> <p>Simulation errors. These are coding errors identified by JOB/SCAN and listed in the SJL output.</p>
10000 >	<div style="border: 1px solid purple; padding: 5px; margin: 5px 0;"> <p> Messages numbered greater than 10000 can be viewed and the severity of these messages can be modified. See - <a href="#">Managing JOB/SCAN's Output Messages</a> for information on Managing Messages in the SJL.</p> </div>

## Product Messages

The product issues two types of message formats:

- Parameterized
- Non-parameterized

Non-parameterized messages contain no replaceable parameters. Parameterized messages contain up to four replaceable parameters (&1, &2, &3, and &4). Because these parameters can vary, depending on system or user input, they have been left unresolved in the text of the error messages listed.



If you receive an error message that is not listed, contact ASG Customer Support.

- [Error Messages from DSS0000 to DSS0100](#)
- [Error Messages from DSS0101 to DSS0300](#)
- [Error Messages from DSS0301 to DSS0500](#)
- [Error Messages from DSS0501 to DSS0700](#)
- [Error Messages from DSS0701 to DSS1000](#)
- [Error Messages from DSS1001 to DSS2000](#)
- [Error Messages from DSS2001 to DSS3000](#)
- [Error Messages from DSS3001 to DSS4000](#)
- [Error Messages from DSS4001 to DSS5000](#)
- [Error Messages from DSS5001 to DSS5600](#)
- [Error Messages from DSS5601 to DSS6001](#)
- [Error Messages from DSS6001 to DSS7001](#)
- [Error Messages from DSS7001 to DSS8000](#)
- [Error Messages from DSS8001 to DSS9000](#)

## DSS0000 ""

**Explanation:** Your password is incorrect or missing. These are the text strings that can appear:

012E XXX Product authorization has expired  
 013E XXX Product authorization invalid  
 014E XXX Product authorization invalid for serial XXXXXX  
 015E XXX Product authorization invalid for model XXXXXX-XXX  
 Product authorization invalid MSU limit exceeded  
 Product authorization invalid MSU invalid on OS390  
 016E XXX Product authorization not found  
 017E XXX Product authorization failure ASGPTBL not found  
 022E XXX Product Evaluation Period has expired  
 032E XXX Temporary Product Authorization has expired  
 033E XXX ASGAPM Ver 2.1 is incompatible with this product  
 034E XXX install ASGAPM version x.x or higher

**Action:** Contact your Product Administrator. For further assistance contact ASG Customer Support.

## DSS0001 INVALID SYNTAX. EXPECTING GROUP LEVEL STMNT

**Explanation:** A Group Level run control statement was expected. The first run control statement encountered must be a Group Level statement. Check the order of the run control statements to verify that a group level statement precedes the detail level statements. A group level statement must begin in column 1, be all upper case letters, and must be one of the following:

- SYSTEM IDENTIFICATION
- CONTROL TABLES
- INPUT SELECTION

- FILE ALLOCATIONS
- ACCESS CONTROL
- DOCUMENTATION KEYWORDS
- RUN OPTIONS
- REPORT OPTIONS

Action: Contact ASG Customer Support.

#### DSS0002 UNIDENTIFIED GROUP LEVEL STATEMENT

Explanation: The statement starting in column 1 was neither a comment nor a Group Level statement. A group level statement must be one of the following:

- SYSTEM IDENTIFICATION
- CONTROL TABLES
- INPUT SELECTION
- FILE ALLOCATIONS
- ACCESS CONTROL
- DOCUMENTATION KEYWORDS
- RUN OPTIONS
- REPORT OPTIONS

Action: Contact ASG Customer Support.



Group level statements must be entered in upper case letters.

#### DSS0003 INVALID SYNTAX. PARM MUST END WITH =

Explanation: The parameter you entered requires an equal sign.

Action: Correct the statement in error and rerun.

#### DSS0004 xxx IS INCORRECT. PARM MUST BE NUMERIC.

Explanation: A non-numeric parameter was used when a numeric entry was required.

Action: Correct the statement in error and rerun.

#### DSS0005 xxx IS INVALID IN DDNAME DEFINITION

Explanation: Diagnostic message produced when trace is turned on.

Action: Debug trace should only be used at the request of ASG Customer Support.

#### DSS0006 xxx - UNIDENTIFIED DETAIL LEVEL STATEMENT

Explanation: The statement entered was not a Detail Level statement.



Action: A Detail Level statement must follow a Group Level statement and must start in column 2 or greater.



Detail level statements must be entered in upper case letters.

#### DSS0007 PRODUCT AUTHORIZATION WILL EXPIRE IN xxxx DAYS

Explanation: Your password is about to expire.

Action: If you have not received a new password, contact ASG Customer Support.

#### DSS0008 THE PASSWORD TRAILER LIST IS > 16 ENTRIES

Explanation: There are more than 16 five-digit entries following your password.

Action: The password trailer list cannot be greater than 16 entries. Correct the password string and trailer.

#### DSS0009 xxx IS INCORRECT AS USED IN THIS STATEMENT

Explanation: The characters entered are incorrect in this statement. Correct the statement.

Action: The problem may occur when a new release of the programs has been installed and this is your first access of the new modules. If this is the problem, enter JOB/SCAN, select the Processing Variables panel, then exit JOB/SCAN. New values update your User Profile and all subsequent uses of JOB/SCAN should be successful.

#### DSS0010 SAME SOURCE CANNOT BE SPECIFIED TWICE

Explanation: The SOURCE parameter of the SHORT DESCRIPTIONS Run Options statement contains two identical entries, for example, JCL,JCL.

Action: Correct the statement and rerun. Valid entries are JCL and FILE. If both are specified, use parentheses and comma ex. (JCL,FILE).

#### DSS0011 PGM.DDNAME AND STEP.DDNAME ARE MUTUALLY EXCLUSIVE

Explanation: The ENTITY levels PGM.DDNAME and STEP.DDNAME cannot both be coded in the same SHORT DESCRIPTIONS Run Options statement.

Action: Correct the statement and rerun.

#### DSS0012 ONLY ONE CHARACTER ALLOWED IN OVERRIDE PARM

Explanation: Multiple characters were entered in the ORI RUN CONTROL statement.

Action: Only one override character is allowed. The character coded as the Override Indicator highlights the override JCL statements in the Structured JCL Listing (SJL). Any single character can be coded in the statement. JOB/SCAN and DOCU/TEXT automatically highlight override DD statements by replacing the normal slash characters (//) with dollar signs (\$\$) as the statements are printed in the SJL. When the override statement is printed in the PROC following the DD statement to which it applies, the // is replaced by \$/. The \$ is the default ORI but can be replaced

with any character. Do not place delimiters around the character. Note that specifying ORI=/ nullifies the highlighting effect of JOB/SCAN and DOCU/TEXT.

#### DSS0013 INVALID SYNTAX.

Explanation: The syntax of the statement entered is incorrect.  
 Action: Check the product documentation for the correct syntax.

#### DSS0014 DEVICE CODE "" IS RESERVED OR WAS ALREADY SPECIFIED

Explanation: The subparameter in the DEVICE CODES Run Option statement contains a device code that was already specified or is one of the reserved codes.  
 Action: Check Control Table AA and make a correction if necessary. There are ten reserved Device Codes:

- C Cards
- D Direct Access (Disk)
- G Graphics (display)
- K Cartridge Tape
- M Mass Storage
- R Report (Printer)
- T Tape (Reel)
- U Unknown Device
- V VIO (Virtual I/O)
- X Dummy (Null) DEVICE CODE entries add to or override devices entered in Control Table AA

#### DSS0015 ONLY THREE CODES ALLOWED FOR THIS PARM

Explanation: More than three device types were entered in the DEVICE CODES Run Option for either DASD, TAPE, or REPORT.  
 Action: DEVICE CODE entries add to or override device codes entered in Control Table AA.

#### DSS0016 MULTIPLE HYPHENS IN RANGE PARAMETER

Explanation: Multiple hyphens are not allowed in a SELECT=range-range INPUT SELECTION statement.  
 Action: INPUT SELECTION is a Group Level statement. SELECT= is a Detail Level statement whose format is as follows: SELECT=range1-range2 (selects specific jobs in this range ex. JOB1-JOB99) SELECT-T=mask (asterisk can act as wildcard in any position) SELECT=PREFIX= (selects jobs of any length that begin with specified PREFIX. ex. PREFIX=PAY) SELECT=ALL (selects all jobs)

#### DSS0017 USE OF MASK IN RANGE IS ILLEGAL

Explanation: You entered an asterisk (\*) in a SELECT=range-range INPUT SELECTION statement.

Action: Asterisks are not allowed in this type of INPUT SELECTION statement. INPUT SELECTION is a Group Level statement. SELECT= is a Detail Level statement whose format is as follows: SELECT-T=range1-range2 (selects specific jobs in this range ex. JOB1-JOB99) SELECT=mask (asterisk can act as wildcard in any position) SELECT=PREFIX= (selects jobs of any length that begin with specified PREFIX. ex. PREFIX=PAY) SELECT=ALL (selects all jobs)

#### DSS0018 FIRST ITEM OF RANGE > SECOND ITEM

Explanation: The item in range 1 is of a higher collating sequence than that of range 2.

Action: Check for typographical errors.

INPUT SELECTION is a Group Level statement. SELECT= is a Detail Level statement whose format is as follows:

SELECT=range1-range2 (selects specific jobs in this range ex. JOB1-JOB99)

SELECT=mask (asterisk can act as wildcard in any position)

SELECT=PREFIX= (selects jobs of any length that begin with specified PREFIX. ex. PREFIX=PAY)

SELECT=ALL (selects all jobs)

#### DSS0019 xxx IS INVALID IN SELECT STRING

Explanation: You entered invalid characters in the SELECT statement.

Action: Correct the statement in error and rerun. INPUT SELECTION is a Group Level statement. SELECT= is a Detail Level statement whose format is as follows:

SELECT=range1-range2 (selects specific jobs in this range ex. JOB1-JOB99)

SELECT=mask (asterisk can act as wildcard in any position)

SELECT=PREFIX= (selects jobs of any length that begin with specified PREFIX. ex. PREFIX=PAY)

SELECT=ALL (selects all jobs)

#### DSS0020 EXPECTED VALUE FOR xxx PARM IS NOT PRESENT

Explanation: You entered a value that is not valid for this parameter.

Action: Check the product documentation and correct the statement in error.

#### DSS0022 INVALID REQUEST FOR USER TEXT ENTITY LEVEL

Explanation: The TEXTENTITY level requested is not valid for this report.

Action: Refer to the Report Options section for more information. User Text may be included in reports at TEXTENTITY levels indicated in the following table:

Report # & Name	JOB	PROC	PGM	FILE Level
3. Structured JCL Listing	x	x	x	x
4. Data Set XREF				x
6. Program XREF			x	
7. PROC XREF		x		
8. I/O Listing	x	x	x	
9. Flowcharts	x	x	x	
10. Job Documentation	x	x	x	
11. Tape Usage Analysis	x			x
14. Report Analysis	x			x

**DSS0023 PASSWORD MISSING FROM membername**

**Explanation:** The PARMLIB member "membername" does not contain a password.

**Action:** Make sure the DDNAME JLRCL library is allocated and contains the correct password and access control statements.

**DSS0024 xxx REQUESTED BUT NOT AUTHORIZED**

**Explanation:** The report you requested is not produced by the modules that you have available.

**Action:** See the Report Summary Table in the Analysis Documents Reports or Operations Documents Reports in ASG-DOCU/TEXT's User's guide. If you think you have received this message in error, contact ASG Customer Support.

ANALYSIS DOCUMENTS	OPERATIONS DOCUMENTS
1. Options in Effect	1. Options in Effect
2. Table of Contents	3. Structured JCL Listing

## ANALYSIS DOCUMENTS

## OPERATIONS DOCUMENTS

3. Structured JCL Listing	10. Job Documentation
4. Data Set XREF	11. Tape Usage Analysis
5. Report XREF	12. Tape Pull List
6. Program XREF	13. Tape VOLSER XREF
7. PROC XREF	14. Report Analysis
8. I/O Listing	
9. Flowcharts	
15. Index	

---

## DSS0025 NUMBER OF KEYWORDS EXCEEDS LIMIT

Explanation: Only 16 keywords are allowed in the Job Documentation report.

Action: Correct the statement and rerun. DOCUMENTATION KEYWORDS is the Group Level statement. USER TEXT= is the Detail Level statement. USER TEXT='keyword1', 'keyword2', 'keyword $n$ ' is the format. Up to 16 'keywords' are allowed.

## DSS0026 JLRCL MEMBER membername NOT FOUND

Explanation: The specified member was not found in the PARMLIB library.

Action: Verify that DDNAME JLRCL specifies the correct OPTIONS Library and that the member exists. The DOCU/TEXT and JOB/SCAN controlling members must reside in one or more libraries allocated to DDNAME JLRCL. A LOGON PROC, CLIST, and/or Batch PROC are set up at the time of installation to allocate JLRCL.

## DSS0027 THE PRECEDING TABLE ENTRY IS UNIDENTIFIED

Explanation: This is an internal error.

Action: Often the problem is that the Control Table entries do not begin in column 6. If that is not the problem, contact ASG Customer Support.

## DSS0028 xxx INVALID OUTSIDE membername - IGNORED

Explanation: The detail statement specified is only valid when placed in the PARMLIB library member "membername".

Action: Contact ASG Customer Support.

## DSS0029 FIRST KEYWORD NOT COMMAND, PANEL, OR PROGRAM

- Explanation: The first keyword in a USER FUNCTION detail statement must be COMMAND, PANEL, or PROGRAM.
- Action: The USER FUNCTION is a Detail Level statement located under the ACCESS CONTROL Group Level statement. The USER FUNCTION statement designates access to a specific User Function accessed directly from the DOCU/TEXT or JOB/SCAN Primary Menu. The format is:

```
ACCESS CONTROL PGM=
USER FUNCTION =PANEL=
CMD=
```

where:

PGM= specifies the program to be invoked whenever this User Function is selected from the Primary Menu

PANEL= specifies the name of the additional user selection panel to be displayed

CMD= specifies the command to be executed whenever the User Function is selected Other parameters may follow and are optional.

#### DSS0030 "" IS AN INVALID REMOTE DESTINATION

- Explanation: The value coded for the DEST parameter in the File Allocations statement is not valid for this system.
- Action: Contact ASG Customer Support.

#### DSS0031 "" PARAMETER NO LONGER SUPPORTED - IGNORED

- Explanation: The parameter specified is no longer supported by DOCU/TEXT or JOB/SCAN.
- Action: It is ignored for compatibility.

#### DSS0032 APPLICATION ACCESS DENIED BY INITIALIZATION USER EXIT

- Explanation: The initialization user exit has denied access to DOCU/TEXT and/or JOB/SCAN.
- Action: Contact your Product Administrator for assistance.

#### DSS0033 DOCU/TEXT - JOB/SCAN ALREADY IN USE, TRY LATER

- Explanation: DOCU/TEXT OR JOB/SCAN is currently running in your address space.
- Action: Only one execution of either DOCU/TEXT or JOB/SCAN is allowed per address space.

#### DSS0034 DEVICE CODE "" IS NOW RESERVED - PLEASE REMOVE

- Explanation: The device code entered in the DEVICE CODES Run Option is now a reserved code. "M" is the reserved code for mass storage devices, "V" is the device code for virtual I/O (VIO) devices, and "K" is the device code for cartridge tape devices.
- Action: Remove the device code indicated in the message from the DEVICE CODES Run Option statement.

#### DSS0035 DEVICE CODE "" IS NOT AN ALPHABETIC (A-Z) CHARACTER

Explanation: The device code entered in the DEVICE CODES Run Option parameter was not an alphabetic character.

Action: Enter a letter from A to Z.

#### DSS0036 UNABLE TO ATTACH D00YCAST SUBTASK - RC xxxx

Explanation: This message is usually the result of an installation error. Module D00YCAST is attached at initialization and the ATTACH macro issued the return code of xxxx.

Action: Check your JOBLIB/STEPLIB concatenation and make sure the library containing D00YCAST is specified.

#### DSS0037 D00YCAST SUBTASK UNEXPECTED RETURN CODE - xxxxxxxx

Explanation: Internal error.

Action: Contact ASG Customer Support.

#### DSS0038 ACCESS CONTROL "AMD" STATEMENTS EXCEEDED MAXIMUM

Explanation: Sixteen "ACCESS MULTIPLE DOCUMENTATION" (AMD) ACCESS CONTROL statements have been processed DOCU/TEXT ignores all but the first sixteen statements.

Action: Contact ASG Customer Support.

#### DSS0039 JOB LIMIT AMOUNT " " EXCEEDS MAXIMUM (99998)

Explanation: The maximum amount for the JOB LIMIT Run Option is 99998 for both batch and foreground processing.

Action: A value of zero (0) allows an unlimited number of jobs.

#### DSS0040 INVALID INFO/X DB2 SUBSYSTEM NAME SPECIFIED: xxxx

Explanation: Invalid DB2 Subsystem Name specified on the TYPERUN statement for INFO/X.

Action: Refer to the IBM JCL Reference Manual for correct DB2 subsystem name syntax or to your systems programmer for the correct subsystem name at your site.

#### DSS0041 INVALID INFO/X xxxxx PARM: xxxxx

Explanation: Invalid INFO/X TYPERUN parameter specified.

Action: Check the INFO/X Administration Guide for valid parameters.

#### DSS0042 ERROR IN INFO/X MODULE: xxxx

Explanation: An error occurred when executing the INFO/X program module specified in the error message.

Action: Contact ASG Customer Support.

#### DSS0043 BYPASSING UPDATE OF INFO/X DB2 REPOSITORY

Explanation: An advisory message is issued when the REPOSITORY=PC TYPERUN parameter is specified.

Action: No attempt will be made to update a mainframe INFO/X repository.

#### DSS0044 XREF FACILITY/TYPERUN PARM MISMATCH. USING xxxx

- Explanation: An advisory message is issued when there is a product mismatch specified on ACCESS CONTROL and RUN CONTROL statements.
- Action: If running "INFO/X", you should specify "INFO/X" on both. If running "OLX", "OLX" should be specified on both. The error message states which product is taken as the default.

**DSS0045 INFO/X ERROR ALLOCATING xxxxxxxx WORK FILE**

- Explanation: An error message is issued when INFO/X cannot allocate the JLFOLT, JLFOLT2, JLCRC, etc, work files.
- Action: Check the INFO/X batch JCL or PARMLIB to ensure proper file allocation information.

**DSS0046 INFO/X UNABLE TO CONNECT TO DB2 SUBSYSTEM: xxxxx**

- Explanation: An error message is issued when INFO/X cannot connect to the DB2 Subsystem as specified on the TYPERUN Run Option statement.
- Action: The subsystem name may not be currently active or an incorrect subsystem name was specified.

**DSS0047 INFO/X UNABLE TO OPEN PLAN: xxxxxxxx**

- Explanation: An error message is issued when INFO/X cannot open the DB2 plan.
- Action: Ensure that a BIND was performed during the INFO/X installation process and that you have proper DB2 authorization to execute the plan.

**DSS0048 INFO/X UNABLE TO CLOSE PLAN: xxxxxxxx**

- Explanation: An error message is issued when INFO/X cannot close the DB2 plan. DSS0049 INFO/X UNABLE TO DISCONNECT FROM DB2 SUBSYSTEM: xxxxx
- Action: An error message is issued when INFO/X cannot disconnect from the DB2 Subsystem.

**DSS0050 "text"**

- Explanation: Diagnostic message produced as a result of a SNAP request.
- Action: This request should be made only under the direction of ASG Customer Support.

**DSS0051 USERTEXT MEMBER NOT FOUND FOR DD xxxxxxxx.**

- Explanation: The USERTEXT member described by DD named xxxxxxxx was not found.
- Action: Contact ASG Customer Support.

**DSS0052 USERTEXT MEMBER xxxxxxxx NOT FOUND.**

- Explanation: The USERTEXT member named xxxxxxxx was not found in the dataset.
- Action: Contact ASG Customer Support.

**DSS0053 PLAN-LIST TABLE OVERFLOW.**

- Explanation: The number of jobs being updated has exceeded the capacity of the table containing PLAN data.



Action: The INFO/X update run should be rerun specifying a fewer number of jobs using the START= and COUNT= parameters.

#### DSS0054 PLAN-DATA TABLE OVERFLOW.

Explanation: The number of jobs being updated has exceeded the capacity of the table containing PLAN data.

Action: The INFO/X update run should be rerun specifying a fewer number of jobs using the START= and COUNT= parameters.

#### DSS0055 VERSION MISMATCH BETWEEN ANALYSIS AND UPDATE PROGRAM

Explanation: The INFO/X Transaction File (JFOLT) created by the TYPERUN PARSEONLY option shows a different version level than that expected by the TYPERUN IMPORT option.

Action: Rerun the TYPERUN PARSEONLY option with the matching version of INFO/X.

#### DSS0056 VALUE REPOSITORY MISMATCH

Explanation: The INFO/X repository read is not VERSION 2.

Action: See your INFO/X administrator.

#### DSS0057 UNABLE TO CREATE CRC FOR MEMBER " "

Explanation: INFO/X was unable to create a CRC value for the member shown.

Action: See preceding associated message that indicates the reason for the failure. For example, an allocation or read failure.

#### DSS0058 " text from DB2 "

Explanation: When INFO/X makes an SQL call which gets a non-zero return code, then DB2 formats the error message and it is provided to the INFO/X user via this message.

Action: Contact ASG Customer Support.

#### DSS0060 NO PARAMETERS WERE PASSED TO D06YSTUT - PROCESSING TERMINATED.

Explanation: The Standards Prompter was invoked incorrectly and could not run. DSS0061 NOT A VALID JOB/SCAN ENVIRONMENT - PROCESSING TERMINATED.

Action: The Standards Prompter was invoked incorrectly and could not run.

#### DSS0062 RC=XXXX AT OFFSET XXXXX IN D06YSTUT (MM/DD/YY HH:MM).

Explanation: The Standards Prompter received an error code from an ISPF service.

Action: Print the screen containing the error message and contact ASG Customer Support.

#### DSS0066 tapemgr2 SPECIFIED - tapemgr1 ALREADY SPECIFIED

Explanation: Two tape management interfaces were specified in the Program PRODUCT INTERFACE Run Option.

Action: Only one tape management interface is allowed (CA1/CA1R5/RMM/ZARA).

**DSS0068 EXCESSIVE NUMBER OF PARAMETERS IN THIS STATEMENT**

Explanation: More parameters were specified than are allowed for this Option statement.

Action: Contact ASG Customer Support.

**DSS0070 GROUP ID SPECIFIED WITHOUT A CORRESPONDING USER ID**

Explanation: The GROUP ID positional parameter must have a corresponding USER ID parameter coded in the SECURITY EXIT Run Option statement.

Action: Contact ASG Customer Support.

**DSS0091 NO MEMBERS SELECTED FOR UPDATE**

Explanation: The Automatic Incremental Update for either Panvalet or Librarian did not find any JCL members which required updating.

Action: Contact ASG Customer Support.

**DSS0092 INVALID DSNAME IN INPUT SELECTION**

Explanation: An invalid dataset name was specified on an INPUT SELECTION statement.

Action: This message will be followed by a DSS0093 message which will identify the specific invalid dataset name.

**DSS0093 DSN=dsname**

Explanation: "dsname" is an invalid dataset name on an INPUT SELECTION statement.

Action: This message will be preceded by a DSS0092 message.

**DSS0094 INPUT SELECTION DDNAME "ddname" NOT FOUND**

Explanation: The DD named "ddname" was not found in the JCL for the job.

Action: When the DDNAME= parameter is specified in an INPUT SELECTION statement, a corresponding DD statement must be specified in the JCL for the job.

**DSS0096 ERROR FOUND IN RCL WHICH PROHIBITS EXECUTION**

Explanation: An error was encountered in the Run Control statements which prohibits further execution.

Action: Correct the error and rerun.

**DSS0097 INTERNAL MODULE SYNCH ERROR - CONTACT VENDOR**

Explanation: DOCU/TEXT has encountered an internal subroutine error.

Action: Contact ASG Customer Support.

**DSS0098 JOB/SCAN USAGE NOT AUTHORIZED**

Explanation: Your password is incorrect or you are not authorized to use JOB/SCAN.

Action: Contact ASG Customer Support.

**DSS0099 ERROR FOUND WHICH PROHIBITS EXECUTION**

Explanation: The error preceding this statement prohibits further execution by DOCU/TEXT.

Action: Correct the error and rerun.

#### DSS0101 INVALID CARRIAGE CONTROL xxx RECEIVED FROM xxx

Explanation: This message is issued when the reporting subroutine has received a print line with an invalid carriage control character.

Action: If the 'CTLCHAR' keyword is coded in the TEXTSOURCE parameter of the Job Documentation (OJD) Report Option and there is not a valid carriage control character in the User Text, this message is issued. If this is not the case, it is an internal error. In this case contact ASG Customer Support.

#### DSS0102 SYSTEM SUMMARY REPORT INACCESSIBLE

Explanation: DDname JLSSR cannot be opened. This could be an installation problem.

Action: Refer to [Installation](#) for procedures to follow when assigning JLSSR. When setting up the ISPF environment, JLSSR is usually allocated to your terminal and gives error listings. JSLLR is dynamically allocated when running JOB/SCAN in the foreground (edit macros or JOB/SCAN ISPF panels) and need not be pre-allocated. When running JOB/SCAN in batch, JSLLR must be allocated via a DD statement pointing to a SYSOUT class or a data set.

#### DSS0103 CANNOT OPEN REPORT DDNAME xxx

Explanation: The ddname specified cannot be opened.

Action: Check the TSO Hardcopy specifications, such as CLASS, on the JOBS STATEMENT AND TSO OUTPUT PARMS panel. Alternatively, verify the FILE ALLOCATIONS statement in the PARMLIB or in the batch PROC.

#### DSS0104 LINES LEFT AFTER PRINT IS xxx

Explanation: Diagnostic message produced as a result of a debug request.

Action: Debug should only be used at the request of ASG Customer Support.

#### DSS0105 OPEN FAILED FOR USER TEXT DDNAME xxx - PROGRAM xxx

Explanation: The specified ddname for the User Text library cannot be opened.

Action: Verify that the specific Report Option requesting User Text data and the FILE ALLOCATIONS statement coded in JSOPT02 (or equivalent member) designating the User Text library, are both present and agree.

#### DSS0106 MEMBER xxxxxxxxxx NOT FOUND IN TEXTSOURCE LIBRARY

Explanation: The JOBNAME, PROCNAME, or PROGRAM name found in the JCL being documented is not contained in the specific User Text library.

Action: This error most often occurs when you do not have User Text members but the User Text option is enabled. TEXTENTITY parameters of various REPORT OPTIONS enable User Text. Valid sub-parameters of TEXTENTITY are JOB,PROC,PGM, and FILE. Search the REPORT OPTION section of JSOPT02 (or equivalent member) for any unwanted TEXTENTITY= parameters or sub-

parameters and disable them. For example: to eliminate the search for FILE level members but to allow the search for PROC level members when including User Text in the SJL Report, change:

REPORT OPTIONS SJL =TEXTSOURCE=FILE,TEXTENTITY=(JOB,PROC,PGM,FILE) to:

REPORT OPTIONS SJL =TEXTSOURCE=FILE,TEXTENTITY=(JOB,PROC,PGM)

Reports that may include the TEXTENETITY parameter are:

3. Structured JCL Listing
4. Data Set XREF
6. Program XREF
7. PROC XREF
8. I/O Listing
9. Flowcharts
10. Job Documentation
11. Tape Usage Analysis
14. Report Analysis

#### DSS0110 JES2 JOBPARM PROCLIB xxxxxxxx WAS NOT FOUND

Explanation: The DDNAME coded on the PROCLIB parameter of the JOBPARM statement was not found.

Action: If a JIIPA DD was allocated, those libraries will be searched.

#### DSS0115 JES3 MAIN PROC xx WAS NOT FOUND

Explanation: The DDNAME coded on the PROC parameter of the MAIN statement was not found.

Action: Either the characters have been coded incorrectly or the JIIPAx was not allocated.

#### DSS0120 INVALID SUBROUTINE CODE xxx in xxx STATEMENT

Explanation: The control statement to debug - snap, trace, or stats - specified an incorrect program code.

Action: This is an internal message. Contact ASG Customer Support.

#### DSS0121 INVALID SERVICE CODE - xxx

Explanation: The process requested in the debug request was other than snap, trace, or stats.

Action: This is an internal message. Contact ASG Customer Support.

#### DSS0122 UNKNOWN PROCESSOR CODE xxx

Explanation: The debug request included a processor code not in the program table.

Action: This is an internal message. Contact ASG Customer Support.

#### DSS0125 xxxxxxxxxxxxxxxxxxxxxxxx TABLE ENTRIES EXCEEDED

Explanation: An internal table was has been exceeded.

Action: Contact ASG Customer Support.

**DSS0172 MISSING OR INVALID TYPE xxx**

Explanation: The value ENTYPE returned from a VSAM catalog access is a value not processed by the system.  
Action: Contact ASG Customer Support.

**DSS0173 NO DATA NAME FOR CLUSTER xxx**

Explanation: The return value for a cluster did not include the name of the data component, which is required to determine the dataset characteristics.  
Action: Contact ASG Customer Support.

**DSS0174 DATA ENTRY xxx NOT FOUND**

Explanation: The VSAM Catalog Access could not find the data component for a cluster when determining dataset characteristics.  
Action: Contact ASG Customer Support.

**DSS0175 ENTYPE IS xxx INSTEAD OF D**

Explanation: The VSAM catalog access for the data component of a cluster returned the entry type specified.  
Action: Contact ASG Customer Support.

**DSS0176 UNEXPECTED RC rc-crs FROM CATALOG xxx**

Explanation: The return code from VSAM catalog access was not one of the expected values.  
Action: See IBM message IDC3009I for return code (rc) and reason code (crs) descriptions.

**DSS0177 CODE xxx FROM GDG READ FOR xxx**

Explanation: A re-read for a GDG record in the VSAM catalog failed.  
Action: Contact ASG Customer Support.

**DSS0182 BAD QUOTED STRING AT COL CC**

Explanation: The end of a line from the rules file was found prior to the end a quoted string. "CC" indicates the last column scanned.  
Action: Check for mismatched delimiting quotes, or incorrect usage of double quotes to represent quotes in the string.

**DSS0183 EXPECTED AN OPERATOR; FOUND xxxxxx**

Explanation: A valid operator was expected at the location being examined. Instead, the string 'xxxxxx' was found.  
Action: String 'xxxxxx' may be a misspelling of the intended operator or the operator may have been omitted entirely.

**DSS0184 OPERATOR 'xxxxxx' IN WRONG SEQUENCE**

Explanation: 'xxxxxx', although a recognized operator, may not appear in this location in the command.

Action: Check for proper ordering of clauses - for example, a "TO" before a "FROM" on a "CHG" causes this error.

#### DSS0185 'xxxxxx' IS NOT A VALID 'ON' OPERAND

Explanation: Only certain values may be used as "ON" operands.

Action: See the product documentation for a current list. 'xxxxxxx' was not one of these specific values. Site-defined non-IBM-standard JES statement types cannot be accessed. Typical ON operands are JOB, EXEC, DD, EXPGM, EXPROC, JES, JES2, JES3, jesstatementname (ex. JOBPARM), etc.

#### DSS0186 TOO MANY UNQUOTED EQUAL SIGNS - 'xxxxxxxxxxxx'

Explanation: More than one unquoted equal sign was found in an IF, AND or OR clause. For example, "IF A=B=C" could cause this error.

Action: If the second equal sign is part of the VALUE of the parameter, enclose it in quotes. 'xxxxxxxxxxxx' is the part of the command that contains this error.

#### DSS0187 EXPECTED EQUAL SIGN NOT FOUND - 'xxxxxxxxxxxx'

Explanation: There was no unquoted equal sign in the portion of the command identified by 'xxxxxxxxxxxx', although one was required by the syntax rules.

Action: This might be caused by omitting or mis-keying an equal sign in an "if" clause - "IF JOB=PJGL0050", instead of "IF JOB=PJGL0050", for example.

#### DSS0188 MORE THAN ONE 'ON' ACTIVE

Explanation: Within the current "group" (all statements between "END"s), two or more "ON" clauses were found.

Action: One and only one "ON" may be active at a time. Possible causes include a missing "END" statement.

#### DSS0189 MORE THAN ONE 'IF' ACTIVE

Explanation: Within the current "group" (all statements between "END"s), two or more "IF" clauses were found.

Action: Only one "IF" may be active at a time. Possible causes include a missing "END" statement.

#### DSS0190 'ON' REQUIRED BEFORE xxxxxx

Explanation: An attempt was made to perform an action (CHG, ADD, DEL, or STRCHG) when no "ON" was yet active within the "group" (all statements between "END"s).

Action: An "ON" must be present before the action may be specified. The action requested is 'xxxxxx'. The ON Selector determines on which kind of JCL statement the change is to occur, ex. ON DD.

#### DSS0192 DFSMS RETURNED CODE xxx, REASON xxx

Explanation: The MVS/DFP DFSMS subsystem has returned an unexpected return code or an unexpected reason code.

Action: Refer to the IBM "DFP: DIAGNOSIS REFERENCE" manual for the meaning of this return code and reason code. See your storage administrator or systems programmer for assistance.

#### DSS0193 INTERNAL LOGIC ERROR - BAD TABLES

Explanation: The internal tables required by the product are out of synchronization with each other.

Action: Contact ASG Customer Support.

#### DSS0194 # OF WILD CARDS (\*'s or %'s) NOT SAME IN FROM AND TO

Explanation: If any wild card characters are used in the "TO" operand, exactly the same number of wild card characters are required in the "FROM" operand.

Action: Contact ASG Customer Support.

#### DSS0195 NO OPERAND SPECIFIED FOR DEL COMMAND

Explanation: A del command was encountered without an operand.

Action: An operand is required after a del command in order to know what is to be deleted.

#### DSS0196 MISMATCHED PARENTHESES IN EXPRESSION

Explanation: When the end of an expression was found, the number of logical open parentheses was not equal to the number of logical close parentheses.

Action: If the mismatch is not obvious, it may be caused by failure to properly quote a parameter value that itself contains parentheses.

#### DSS0197 CHG \* IS THE ONLY VALID WILDCARD CHG SYNTAX

Explanation: A wildcard character was found in the operand of the "CHG" verb, but the operand was not just a single asterisk.

Action: For example, "CHG K\*" would cause this syntax error. The only valid syntax for the "CHG" command that involves a wildcard is "CHG \*".

#### DSS0198 NO \*'S ALLOWED FOR ADD

Explanation: An attempt was made to "ADD" a keyword that contained an asterisk in its name. For example, "ADD B\*=1".

Action: This cannot be properly interpreted, and is therefore disallowed by the syntax checker.

#### DSS0199 IN 'DEL', \*'s ALLOWED ONLY RIGHT OF '='

Explanation: For a "DEL" command, a wildcard character (an asterisk) was found left of an equal sign - for example, "DEL X\*=N".

Action: This syntax is not allowed due to the potential for unintended results.

#### DSS0200 TWO \*'s IN A ROW ARE INVALID

Explanation: When specifying the operand of a "FROM" preposition there is never a need to place two wildcard characters (\*'s) consecutively.

Action: Such a configuration can always be replaced by a single "\*".

**DSS0201 GETMAIN/FREEMAIN FAILED FOR NESTING PARENS**

- Explanation: An attempt to acquire or release storage has failed.
- Action: Try re-executing the job since this error could be caused by a temporary storage shortage. If the problem persists, contact ASG Customer Support.

**DSS0202 OPERAND OF ADD MUST HAVE AN EQUAL SIGN**

- Explanation: An "ADD" was found whose operand didn't include a value specification.
- Action: For example, "ADD G" is not valid because no value for "G" was given.

**DSS0203 '=' TOO FAR RIGHT IN OP OF ADD/DEL**

- Explanation: Although an equal sign was located in the operand of an "ADD" or "DEL" command, it was so far to the right that the "keyword" portion of the operand would have been longer than the allowed maximum.
- Action: The keyword portion of the operand may not exceed 20 characters.

**DSS0204 KEYWORD IN IMPLICIT DEL KYWD=\* TOO LONG**

- Explanation: A "DEL" command was found with no equal sign in its operand.
- Action: This is ordinarily interpreted as a "DEL" for all values of the keyword, that is, "DEL kywd" is interpreted as "DEL kywd=\*". In this case however the "kywd" was longer than the allowed maximum, so the command cannot be interpreted. The keyword portion of the operand may not exceed 20 characters.

**DSS0205 UNABLE TO COMPLETE CHG, NEW VALUE TOO LONG:**

- Explanation: This error indicates that an attempted "CHG" will not be completely successful, because the receiving field for the new value is not long enough to contain the value that would have been generated.
- Action: The result is truncated to the maximum length of the receiving field. This error is always accompanied by errors DSS0213 through DSS0220, which give the pertinent details.

**DSS0206 SKIPPING TO 'END', THEN SYNTAX-CHECKING ONLY**

- Explanation: A serious error has caused the by-passing of all subsequent portions of the current "group" (all statements between "ENDS"s).
- Action: In addition, due to the serious error indicated, only syntax-checking will be performed on subsequent groups and no changes will be performed in this run.

**DSS0207 'TO' SAME AS 'FROM' - NO CHANGE WILL OCCUR**

- Explanation: A "CHG" was examined for which the operand of the "TO" was identical to the "FROM".
- Action: This is flagged with an advisory message so that the user may decide whether a typographical error was made.

**DSS0208 'END' MISSING IN LAST REQUEST - ASSUMED**



Explanation: The last group in the input stream did not end with an "END" statement.  
 Action: One has been assumed in order to make the command(s) syntactically correct.

#### DSS0209 NO SYNTAX ERRORS IN THESE COMMANDS

Explanation: The commands specified by the user have been successfully processed by the syntax checker, with no errors of severity greater than zero.  
 Action: This message indicates that the indicated actions will be performed.

#### DSS0210 JVER COMPLETED

Explanation: The JVER Edit Macro completed successfully.  
 Action: Error messages follow any commands containing errors.

#### DSS0212 NO OPERAND SPECIFIED FOR ANNOTATE COMMAND

Explanation: An operand must be specified when using the ANNOTATE command.  
 Action: Contact ASG Customer Support.

#### DSS0213 JOB NAME: JJJJJJJJ ,MEMBER NAME: MMMMMMMMMM

Explanation: The job name and member name for which error DSS0205 was issued.  
 Action: Contact ASG Customer Support.

#### DSS0214 KEYWORD: KKKKKKKKKKKKKKKKKKKKK

Explanation: The keyword for which error DSS0205 was issued.  
 Action: Contact ASG Customer Support.

#### DSS0215 ORIGINAL VALUE:

Explanation: Header for error DSS0205 details.  
 Action: Contact ASG Customer Support.

#### DSS0216 VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV

Explanation: The original value from which an attempt to change resulted in error DSS0205.  
 Action: Contact ASG Customer Support.

#### DSS0217 'FROM' PATTERN:

Explanation: Header for error DSS0205 details.  
 Action: Contact ASG Customer Support.

#### DSS0218 PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP

Explanation: The original pattern under which an attempt to change resulted in error DSS0205.  
 Action: Contact ASG Customer Support.

#### DSS0219 'TO' PATTERN:

Explanation: Header for error DSS0205 details.



**DSS0228 'ANNOTATE xxxxxxxx' MUST PRECEDE FIRST 'ON'**

Explanation: The ANNOTATE command must be coded prior to the first 'ON' Selector JCL Change command.

Action: Contact ASG Customer Support.

**DSS0229 FIRST PARAMETER NO LONGER FITS ON LINE**

Explanation: Reformatter has attempted to adjust a quoted parameter that is continued.

Action: The quoted parameter has either not met the MVS requirement of column 71 being non-blank and the continuation starting in column 16, or the adjustment caused a quote to be moved to column 71.

**DSS0230 JCL CHANGE INTERNAL ERROR: RC=XYY -CONTACT ASG SUPPORT**

Explanation: A general error message from CHANGE processing.

Action: Contact ASG Customer Support.

**DSS0231 UNABLE TO ATTACH D08YVAL SUBTASK - RC xxxx**

Explanation: An attempt to attach the JCL Change/Reformatter validation subtask, D08YVAL, has failed.

Action: Contact ASG Customer Support.

**DSS0232 D08YVAL SUBTASK UNEXPECTED RETURN CODE - xxxxxxxx**

Explanation: The JCL Change/Reformatter validation subtask has returned an unexpected return code. xxxxxxxx is the code returned.

Action: Contact ASG Customer Support.

**DSS0233 VALIDATION SUBTASK MODULE D08YVAL NOT FOUND**

Explanation: The JCL Change/Reformatter validation subtask module, D08YVAL, could not be found.

Action: Check the JOBLIB/STEPLIB concatenation or JLLOD library (if not running APF-authorized batch).

**DSS0234 'xxxx' IS NOT VALID WITH THIS 'xxxx' COMMAND**

Explanation: An invalid parameter was found in the specified command.

Action: See [Reference Information](#) for valid parameters for Reformatter commands.

**DSS0235 'nnnn' IS NOT A VALID NUMBER FOR 'xxxx'**

Explanation: The number 'nnnn' either contains invalid digits or is out of the valid range for the Reformatter verb or parameter 'xxxx'.

Action: See the documentation for information on what numbers are valid with the command.

**DSS0236 'xxxxxx' NOT ALLOWED WITHIN 'IF'**

Explanation: The verb shown in the error message may not be used within 'IF' logic.

Action: Contact ASG Customer Support.

**DSS0237 KEYWORD TOO LONG**

Explanation: While processing a 'CHG' command, a keyword was deleted which exceeded the maximum allowable length.

Action: This is probably due to a typographic error in the input command(s).

**DSS0238 'xxxx' NOT ALLOWED WITHIN 'ON xxxxxx'**

Explanation: The 'xxxx' verb may not be used within the specified type of 'ON'.

Action: Contact ASG Customer Support.

**DSS0239 INVALID PREPOSITION 'xxxx'**

Explanation: 'xxxx' is not a valid preposition in the context of the command in which it appears.

Action: Either the command is in the wrong 'ON' section, the specified preposition is not valid with the other parameters in the command, or the preposition was specified incorrectly.

**DSS0240 UNEXPECTED YMS RC nn; PROGRAM ENDED**

Explanation: Internal error.

Action: Contact ASG Customer Support.

**DSS0241 INVALID REQUEST xxx TO ICAT PROCESSOR**

Explanation: The internal catalog processor received an invalid request. This is an internal error.

Action: Contact ASG Customer Support.

**DSS0243 NRF IN VTOC READ FOR PERM**

Explanation: This is an internal error.

Action: Contact ASG Customer Support.

**DSS0244 NRF IN VTOC FOR TEMP DSN**

Explanation: This is an internal error.

Action: Contact ASG Customer Support.

**DSS0247 'ON NONJCL' PERMITS 'STRCHG' ONLY**

Explanation: An attempt was made to perform an ADD, CHG, or DEL command on a 'NONJCL' statement type (normally, control card input).

Action: This is not allowed. The only action that JCL Change will permit on control cards is a STRCHG, because control cards are not broken down into individual parameter fields. Only the 80-byte image is available to JCL Change.

**DSS0248 EOF WHILE CHECKING FOR 'MISSING'**

Explanation: A command began with 'IF ..... IS '. JCL Change was attempting to find the word 'MISSING', which would be required in such a case, but instead found the end of the command member.

Action: Probably caused by an entire line or lines being omitted from the command member.

#### DSS0249 'MISSING' INVALID WITH THIS 'ON' CONDITION

Explanation: Only certain 'ON' clauses can be used with 'MISSING' tests.

Action: In particular, only 'ON EXEC', 'ON EXPGM', 'ON EXPROC', and 'ON JOB' are allowed to have 'MISSING' tests subordinate to them.

#### DSS0250 MEMBER xxxxxxxx SUCCESSFULLY CONVERTED - NO CHANGE

Explanation: The automated conversion program that upgrades pre-release-5.6 command members to 5.6 syntax has completed for the member indicated, and no change was required.

Action: Contact ASG Customer Support.

#### DSS0251 MEMBER xxxxxxxx SUCCESSFULLY CONVERTED - CHANGED

Explanation: The automated conversion program that upgrades pre-release-5.6 command members to 5.6 syntax has completed for the member indicated, and the required changes have been made.

Action: Contact ASG Customer Support.

#### DSS0252 CONVERSION COMPLETE

Explanation: The automated conversion program that upgrades pre-release-5.6 command members to 5.6 syntax has completed.

Action: Contact ASG Customer Support.

#### DSS0253 INVALID 'IF MISSING' TEST FOR THIS 'ON'

Explanation: Only certain items may be checked for presence using the 'IF ... IS MISSING' syntax. This table shows what missing tests maybe used.

UNDER	VALID
'ON EXEC'	'IF DD=xxxxxxx IS MISSING'
'ON EXPGM' or 'ON EXPROC'	'IF OUTPUT=xxxxxxx IS MISSING'
'ON JOB'	'IF STEP=xxxxxxx IS MISSING'
	'IF EXPGM=xxxxxxx IS MISSING'
	'IF PGM=xxxxxxx IS MISSING'

UNDER	VALID
	'IF EXPROC=xxxxxxx IS MISSING'
	'IF PROC=xxxxxxx IS MISSING'
	'IF DD=JOB CAT IS MISSING'
	'IF DD=JOB LIB IS MISSING'
	'IF JCL LIB IS MISSING'
	'IF xxxxxxxx IS MISSING'

where xxxxxxxx is any recognized JES2 or JES3 statement type (such as JOBPARM)

#### DSS0254 '=' MUST BE SPECIFIED FOR THIS 'MISSING' TEST

- Explanation:** Most tests for items being 'MISSING' must include the value of the item being tested.
- Action:** For example, it is not valid to say 'IF DD IS MISSING', but it is valid to say 'IF DD=SYSUT1 IS MISSING'. Specific 'MISSING' tests that always require specification of a value include:  
DD OUTPUT STEP EXPGM PGM EXPROC PROC

#### DSS0255 '=' INVALID FOR THIS 'MISSING' TEST

- Explanation:** Some tests for items being 'MISSING' do not allow the specification of particular values for the item being tested.
- Action:** For example, it is not valid to say 'IF JOBPARM=XYZ IS MISSING', but it is valid to say 'IF JOBPARM IS MISSING'. Specific 'MISSING' tests that do not allow specification of values include, JCL LIB and All JES2 and JES3 statement types.

#### DSS0256 NO 'ENDADD' FOUND FOR ADD STMT(S)/LINE(S)

- Explanation:** Every ADD STMT, ADD STMTS, ADD LINE, or ADD LINES command must be delimited by an 'ENDADD' statement.
- Action:** The characters 'ENDADD' must appear in columns 1-6, and the rest of the line must be completely blank. If such a delimiter is not found, this error will occur.

#### DSS0257 THIS 'ON' PERMITS 'STRCHG' ONLY

- Explanation:** Under the clauses 'ON SET', 'ON IF', 'ON ELSE', 'ON ENDIF', and 'ON NEWJCL', the only permitted operation is 'STRCHG'.
- Action:** An attempt was made to perform a CHG, ADD, or DEL under one of these clauses. This is not possible, since these statement types do not follow IBM standard JCL syntax rules and are therefore not broken down into parameter units.

#### DSS0258 AN ERROR WAS FOUND ON THE STATEMENT ABOVE - SEE JLSSR

- Explanation: This message appears after the statement which has an error. The statement is marked with an E.
- Action: The messages pertaining to this statement are in the system diagnostics report (JLSSR).

#### DSS0260 UNKNOWN OPERATOR

- Explanation: A JCL statement did not contain the operation field or a valid JCL verb (JOB, EXEC, DD, etc.) or operator command.
- Action: This is usually the result of an omission or a misspelling.

#### DSS0261 UNEQUAL NUMBER OF RIGHT PARENTHESES

- Explanation: The end of the statement or the keyword was reached and the number of right parentheses exceeded the number of left parentheses.
- Action: Contact ASG Customer Support.

#### DSS0262 OPERAND FIELD DOES NOT TERMINATE IN COMMA OR BLANK

- Explanation: The statement above ends in column 71, is not a comma or a blank, and column 72 is non-blank, indicating a continuation.
- Action: Contact ASG Customer Support.

#### DSS0263 INVALID CONTINUATION SYNTAX

- Explanation: The JCL statement extends past column 72. Make sure that no parameters extend beyond column 72.
- Action: If the statement needs to be continued, code '/' in columns 1 & 2 of the following statement and continue the interrupted portion of the statement in any column from 4 through 16. For example:  
// CONTINUATION CARD

#### DSS0264 EXPECTED CONTINUATION NOT RECEIVED

- Explanation: The previous line of JCL implied that it should be continued on the next by the presence of a comma as the last character or a non-blank in column 72, but the next line contained the start of a new statement.
- Action: Contact ASG Customer Support.

#### DSS0265 MEMBER=mmmmmmmm,JOB=jjjjjjj,SEQ=sssssss

- Explanation: The MEMBER name, JOB name, and SEQUENCE number of the statement in error.
- Action: This message is displayed by CHG/REF module D08YCHPJ after a parse error is detected. It is written to the JLSSR file after the parse error message.

#### DSS0268 'ANNOTATE xxxxxxxx' MUST FOLLOW A VALID 'ON'

- Explanation: The ANNOTATE xxxxxxxx command must be coded after a valid 'ON' Selector JCL Change command.
- Action: Contact ASG Customer Support.

**DSS0270 MEMBER membername NOT PROCESSED**

Explanation: The specified member name was not processed because of a parsing error.

Action: Contact ASG Customer Support.

**DSS0271 MEMBER membername WRITTEN AS RECEIVED**

Explanation: Due to a detected error(s), the specified member was written to the output dataset as it was received.

Action: Contact ASG Customer Support.

**DSS0272 INPUT STREAM CONTAINS NO JOBS**

Explanation: The JCL stream to be changed contains no input.

Action: This may be caused by misspelling the member name, by accessing the wrong input library, or by selecting from a library that contains PROCs whose library type was not specified as PROCLIB. If you are using ISPF JOB/SCAN, you must specify one or more JCL members for processing in the Select Members field or Select List field of the Execution panel. Verify that the library is the correct one by looking at the entry in the Input Lib or List Lib field. To change the Library, overwrite the field. If you are using batch submission, check that INPUT SELECTION statements are entered and spelled correctly. These statement types are: SELECT Specifies JCL member names (ALL selects all members)

EXCLUDE Excludes specific members from among those selected

DSN Identifies library to be accessed.DDNAME Identifies ddname of library to be accessed

SOURCE TYPE Identifies type of library (PAN, LIB, LIB-3, PROC or sequential - Default is PDS)

**Example:**

INPUT SELECTION DSN=SYS2.JOBS SELECT=JOB1-JOB99 Select a range of jobs range1-range2 SELECT=PREFIX=PAY Select all jobs beginning with certain chars

EXCLUDE=JOB50 Select jobs fitting a mask (ex. JOB\*\*UPD) SELECT=ALL Selects all jobs in a system

Repeat the INPUT SELECTION Group Level statement and DSN and SELECT parameters to specify multiple libraries.

**DSS0275 GETMAIN/FREEMAIN FAILED FOR VALUE ROUTINE**

Explanation: An attempt to acquire or release storage has failed.

Action: Try re-executing the job since this error could be caused by a temporary storage shortage. If the problem persists, contact ASG Customer Support.

**DSS0301 CANNOT OBTAIN FIRST BLOCK FOR xxx TABLE**

Explanation: The table processing routine could not obtain storage to initialize the named table.

Action: This is normally an indication of an undersized region. ASG recommends a minimum logon or batch region size of 2048K.

**DSS0302 THE xxx TABLE IS FULL**



Explanation: There is no virtual storage remaining in the region, or the named table has expanded to its specified number of extents.

Action: Make sure your region specification is the largest size available per your site's standard. If it is and you are having this problem, contact ASG Customer Support.

#### DSS0304 OVERFLOW IN TABLE xxx AFTER INSERT

Explanation: Internal error. The pre-computed location for the inserted record was too small for the record size.

Action: Contact ASG Customer Support.

#### DSS0305 SEARCH TABLE xxx AND NO FIT FOR RECORD

Explanation: The preceding table entry is unidentified. This is an internal error.

Action: Contact ASG Customer Support.

#### DSS0306 TABLE xxx HAS ENTRY BUT NO BLOCK POINTER

Explanation: Internal error. One of a set of two pointers is zero.

Action: Contact ASG Customer Support.

#### DSS0307 NO ENTRY ADDRESS FOR ADD IN TABLE xxx

Explanation: An add-record was requested, but no record address was specified. This is an internal error.

Action: Contact ASG Customer Support.

#### DSS0308 ERROR IN TABLE xxx - COMPRESS CHAINING

Explanation: This is an internal error.

Action: Contact ASG Customer Support.

#### DSS0309 TABLE xxx GIVEN NO ENTRY ADDRESS

Explanation: Internal error. A binary search of a keyed table has no key specified on which to match.

Action: Contact ASG Customer Support.

#### DSS0310 TABLE xxx'S BLOCK TOO SMALL FOR ENTRY

Explanation: This is an internal error.

Action: Contact ASG Customer Support.

#### DSS0311 TABLE xxx HAS ZERO LENGTH ENTRY

Explanation: This is an internal error.

Action: Contact ASG Customer Support.

#### DSS0312 VL SYNCH ERROR BACKWARD SEARCH IN xxx

Explanation: This is an internal error.

Action: Contact ASG Customer Support.

**DSS0313 VL SYNCH IN TABLE xxx**

Explanation: This is an internal error.  
Action: Contact ASG Customer Support.

**DSS0314 VL TABLE xxx SORT REQUESTED**

Explanation: This is an internal error.  
Action: Contact ASG Customer Support.

**DSS0315 TABLE xxx UNKEYED - REQUESTED KEYED FUNCTION**

Explanation: A function such as a binary search or add-key-unique was specified on a table which does not define a keylength.  
Action: Contact ASG Customer Support.

**DSS0317 TABLE xxx VL ENTRY NOT MATCHED IN SEARCH**

Explanation: This is an internal error.  
Action: Contact ASG Customer Support.

**DSS0320 D10XFC - xxxxxxxxxx TABLE HAS OVERFLOWED**

Explanation: Internal error -  
Action: Contact ASG Customer Support.

**DSS0323 TAPE VOLUME-SERIAL TABLE OVERFLOW**

Explanation: The number of volume serial numbers for tapes is greater than the DOCU/TEXT maximum of 310.  
Action: Contact ASG Customer Support.

**DSS0325 DUPLICATE CODE "" IN CONTROL TABLE AV - IGNORED**

Explanation: The device code (above) has already been specified in the AV Control Table and is contained in the DOCU/TEXT Flowchart Symbol Table.  
Action: Remove the duplicate codes in Control Table AV.

**DSS0326 FLOWCHART SYMBOL TABLE FULL - AV CODE "" NOT ADDED**

Explanation: The DOCU/TEXT Flowchart Symbol Table can contain nine (9) user-defined device code flowchart symbols specified in Control Table AV. Nine user-defined symbols have already been added to the table and there is no room for any more.  
Action: Remove the symbol for the device code named in the message from Control Table AV.

**DSS0351 REQUEST xxx FROM xxx NOT VALID**

Explanation: Internal - Request from COBOL processor to call interface named an unrecognized processor.

Action: This is an internal message. One possible problem is that ddname JLLOD in the FILEALLOCATIONS statements in PARMLIB points to an incorrect library. Also ensure that all ASG load modules are at the same release level. If this is not the problem, contact ASG Customer Support.

#### DSS0352 PROCESSOR FOR xxx IS NOT LOADED

Explanation: The requested subroutine is not in the composite link and cannot be loaded.

Action: This could be an installation problem. Contact ASG Customer Support.

#### DSS0353 RETURN FROM xxx WAS CODE xxx - CALLED BY xxx

Explanation: Provides a trace of every call through the call interface processor.

Action: This is a diagnostic message produced when trace is turned on. Trace should only be used upon request of ASG Customer Support.

#### DSS0360 D00YAAI RELEASE xxxx - CALLED BY RELEASE yyyy

Explanation: This message is usually the result of an installation error. Module D00YAAI is at release level *xxxx*, and the release level of JOB/SCAN or DOCU/TEXT is *yyyy*.

Action: Review the installation procedure and make sure all load modules are at the same release level. Do not forget that if D00YAAI is APF-authorized (name is in IKJTSOxx member of SYS1.PARMLIB), it must reside in the job pack area (JOBLIB/STEPLIB, LPA, or LNKLSTxx). The datasets named in DD ISPLLIB will not be searched. If you are testing a new release, you can code "NOAUTHFUNCTIONS" in the ISPF Specification Run Option to eliminate this error. When the new release is ready to become production, replace module D00YAAI with the version at the new release level.

#### DSS0361 D00YAAI RECEIVED INVALID REQUEST CODE xx

Explanation: Internal error.

Action: Contact ASG Customer Support.

#### DSS0371 S8A CANNOT MATCH STATUS xxx FOR xxx

Explanation: Internal error. A disposition auditing routine could not determine the resultant disposition of the specified dsname due to incomplete inputs.

Action: One possible problem is that ddname JLLOD is allocated to an incorrect library. Also ensure that all ASG load modules are at the same release level. Alternatively, this problem is sometimes encountered when ISPF split screen mode is engaged if JOB/SCAN is active on one screen and then JEM Edit Macro is invoked on the second screen. If this is not the problem, Contact ASG Customer Support.

#### DSS0373 S8A CASE IS FOR ERROR xxx, DSN xxx

Explanation: A trace of all disposition from the disposition auditing routine was requested.

Action: This is a normal message resulting from a Trace request. Trace should only be used at the request of ASG Customer Support.

**DSS0374 S8A ERROR xxx, DSN xxx**

Explanation: Internal error. A disposition auditing routine could not determine the resultant disposition of the specified dsname due to contradictory inputs.

Action: Contact ASG Customer Support.

**DSS0390 UNABLE TO LOAD MODULE ASASYMBM (FOR SYSTEM SYMBOLS)**

Explanation: The system service routine for substituting system symbols could not be loaded.

Action: Add the load library name that contains the load module named "ASASYMBM" to the JOBLIB/STEPLIB DD concatenation and run the job again.

**DSS0391 D00YSSYM RETURNED UNEXPECTED RETURN CODE - xxxxxxxx**

Explanation: Internal error.

Action: Contact ASG Customer Support.

**DSS0402 xxx IS INVALID REQUEST TO MBR SELECT**

Explanation: This is an internal error. The request code to member select is invalid.

Action: Contact ASG Customer Support.

**DSS0403 x IS INVALID TYPE OF MEMBER SELECTION**

Explanation: This is an internal error. The type of selection specified to member select is invalid.

Action: Contact ASG Customer Support.

**DSS0406 OPEN FAILED FOR INPUT FILE x**

Explanation: The dsname or ddname specified to member select failed open.

Action: A possible cause of this problem is the coding of INPUT SELECTION statements. Statements must be in all upper-case letters.

Example:

INPUT SELECTION Group Level statement - begin in column 1 DSN=SYS2.JOBS Detail Level stmt - begin after column 1

Check that correct library is listed.

SELECT=JOB1-JOB99 Select jobs, ranges, mask, PREFIX= SELECT=PREFIX=PAY SELECT=PAYROL\*\*  
SELECT=JOB123,JOB565 SELECT=ALL Can also select all jobs in the system. Continue Detail Level statements with a space. The continuation line must begin after column 1. Blanks are not allowed within the parameter portion of the detail level statements, except if continued.

If this is not the problem, contact ASG Customer Support.

**DSS0408 MEMBER xxx SELECTED AND NOT FOUND**

Explanation: The specified member was selected by name and could not be found in the specified libraries.

Action: Verify that the correct library containing the member has been specified, using the Library Specification panel.

#### DSS0409 MEMBER xxx ALREADY SELECTED

Explanation: The member has already been selected.

Action: This second selection attempt was by explicit request rather than range or mask.

#### DSS0410 REQUEST CODE xxx INVALID

Explanation: The request block to the dynamic allocation interface routine contained an invalid request - internal error.

Action: Contact ASG Customer Support.

#### DSS0411 DSORG xxx IS AN INVALID VALUE

Explanation: The DSORG in the dynamic allocation request is not one of the combinations allowed.

Action: Refer to [System Administrator Tasks](#) for allowed/valid values.

#### DSS0412 RECFM xxx IS AN INVALID VALUE

Explanation: The RECFM in the dynamic allocation request is not one of the combinations allowed.

Action: Refer to [System Administrator Tasks](#) for allowed/valid values.

#### DSS0413 ALLOC OF xxx FAILED CODE xxx REASON xxx

Explanation: Dynamic allocation of the specified ddname failed. The code value displayed is the Dynamic Allocation Return Code (DARC).

Action: The most common interpretation of this code is displayed as the reason. This could be an installation problem. Also the reports generated may be incomplete. Contact ASG Customer Support. If the REASON given is "SECURITY DENIED", then the CODE value is the return code from the Security Interface Exit and the dataset is deleted from the File Allocation table. If the dataset was part of a concatenation, the rest of the datasets in the concatenation will be allocated and processing will continue.

#### DSS0414 DEALLOC OF xxx FAILED CODE xxx REASON: xxx

Explanation: Dynamic deallocation of the specified ddname failed. The code value displayed is the Dynamic Allocation Return Code (DARC).

Action: The most common interpretation of this code is displayed as the reason.

#### DSS0415 OPEN OUTPUT OF xxx FOUND xxx CONCATENATED

Explanation: Output files cannot be concatenated.

Action: Contact ASG Customer Support.

#### DSS0416 INTERNAL ERROR - CONCAT OF xxx CODE xxx

Explanation: Internal error. Invalid status code in member of concatenation.

Action: Contact ASG Customer Support.

**DSS0417 DDNAME xxx IS ALREADY TABLED AND ALLOCATED**

- Explanation:** A FILE ALLOCATIONS entry in one of the PARMLIB members contains a DDNAME that has been previously allocated by JCL or the TSO Allocate command.
- Action:** This message is also issued when a DDNAME that is normally allocated dynamically by the product was found to be previously allocated. Datasets allocated prior to invocation of the product will be used in lieu of the FILE ALLOCATIONS entry or the dataset specified on the ISPF panel.

**DSS0418 DECONCAT OF xxx FAILED CODE xxx REASON: xxx**

- Explanation:** Dynamic deconcatenation of the specified ddname failed. The code that is displayed is the SVC 99 Dynamic Allocation Return Code (DARC). The most common interpretation of this code is displayed as the reason.
- Action:** Consult IBM's Systems Programmer Library series of manuals.

**DSS0420 AND WAS FOR DSNAME = xxx**

- Explanation:** Specifies the dataset name for the ddname in Error Message DSS0413A.
- Action:** Contact ASG Customer Support.

**DSS0421 OUTADD FAILED FOR DDNAME ddname xxxx yyyy zzzz**

- Explanation:** A Dynamic Output (SVC 109-33) OUTADD request has failed on behalf of the named DDNAME.
- Action:** "xxxx" is the return code from the OUTADD request, "yyyy" is the reason code, and "zzzz" is the text unit key in error.

**DSS0422 OUTADD FAILED FOR OUTPUT name xxxx yyyy**

- Explanation:** A Dynamic Output (SVC 109-33) OUTDEL request has failed on behalf of the named OUTPUT.
- Action:** "xxxx" is the return code from the OUTDEL request, "yyyy" is the reason code, and "zzzz" is the text unit key in error.

**DSS0423 OUTPUT PARAMETERS FOR DDNAME ddname NOT SUPPORTED**

- Explanation:** In the File Allocations statement for DDNAME "ddname", Dynamic Output parameters such as FORMDEF and PAGEDEF were specified.
- Action:** These parameters were not supported until MVS/SP Release 3.1. If you need these parameters, you must code them in an OUTPUT JCL statement and refer to that statement in a JCL DD statement preallocated in your JOB JCL or TSO LOGON procedure.

**DSS0430 SELECTION EXIT - xxx NOT FOUND**

- Explanation:** The user selection exit specified was not found.
- Action:** Check the spelling of the exit name and check the library names allocated to ISPLLIB, JLLD, and STEPLIB. Verify that the exit was linked into one of the load libraries contained in the concatenation lists of the above ddnames. These ddnames may be found in your LOGON PROC,

LOGON CLIST, DOCU/TEXT or JOB/SCAN Invocation CLIST, and/or the JOB/SCAN Edit Macros if you are running online. When running in batch, these allocations are done in your batch JCL, DOCU/TEXT PROC, or JOB/SCAN PROC.

#### DSS0431 RETURN CODE xxx INVALID FROM SELECTION EXIT

- Explanation: The selection exit return code is a value that is negative, greater than 20, or not a multiple of 4.
- Action: Refer to the logic in the selection exit program. The program must return an acceptable value that can be processed by JOB/SCAN. [Input Selection Group](#) describes the Selection Exit Facility and valid values.

#### DSS0432 JOB/SCAN ENDED - SELECTION EXIT RETURN CODE = 20

- Explanation: The JOB/SCAN run has been cancelled by the Selection Exit specified in the INPUT SELECTION statement or in the JEM or JSCAN invocation.
- Action: Contact ASG Customer Support.

#### DSS0434 INPUT FILE OPEN FAILED - ACCESS DENIED BY SECURITY

- Explanation: The input JCL library could not be opened.
- Action: The security system has denied access to the following dataset (see DSS0435).

#### DSS0435 DSN = dsname

- Explanation: The Data Set Name referred to in the previous message.
- Action: Contact ASG Customer Support.

#### DSS0440 UNABLE TO LOAD CONTROL-M MODULE CTMPARM

- Explanation: The PROGRAM PRODUCT INTERFACE (PPI) Run Option indicated that CONTROL-M was active on the system; however, the CONTROL-M module "CTMPARM" could not be loaded.
- Action: Check the STEPLIB or JLLDD DD statements to be sure they specify the CONTROL-M load library. If CONTROL-M is not available at your installation, remove the PPI=CONTROL-M parameter from the PARMLIB.

#### DSS0441 UNABLE TO LOAD CONTROL-M MODULE CTMJSP

- Explanation: The PROGRAM PRODUCT INTERFACE (PPI) Run Option indicated that CONTROL-M was active on the system; however, the CONTROL-M module "CTMJSP" could not be loaded.
- Action: Check the STEPLIB or JLLDD DD statements to be sure they specify the CONTROL-M load library. If CONTROL-M is not available at your installation, remove the PPI=CONTROL-M parameter from the PARMLIB. See the product documentation for more information about the CONTROL-M Interface.

#### DSS0442 CONTROL-M CTMJSP OPCODE "" RETURN CODE "" MEMBER ""

- Explanation: The CONTROL-M Auto Edit Simulation routine has returned an unexpected return code.
- Action: This indicates a problem with CONTROL-M. Operation codes:  
 0 - Initialization call before start of work  
 1 - Before start of new member

2 - Before sending a new line to the internal reader  
 3 - After the end of a member  
 4 - Before completion of work Expected Return Codes:  
 0 - No action taken - send the line  
 4 - Line to send has been processed - send the line  
 8 - Not enough memory (on initialization)  
 12 - Do not send the line due to be sent  
 16 - Cancel the submission of entire member  
 20 - Internal error - cancel submission of entire member See the product documentation for more information about the CONTROL-M Interface.

#### DSS0443 CONTROL-M EXIT 2 OPCODE " " RETURN CODE " "

**Explanation:** The CONTROL-M Exit 2 (job submission exit) routine, CTMX002 has returned an unexpected return code.

**Action:** This indicates a problem with CONTROL-M. Operation codes:  
 0 - Initialization call before start of work  
 1 - Before start of new member  
 2 - Before sending a new line to the internal reader  
 3 - After the end of a member  
 4 - Before completion of work  
 5 - With a line which is not to be sent for some reason. The exit routine may force sending of the line.  
 6 - Call from CTMJSP to check whether it is authorized to access the JCL library. Expected Return Codes:  
 0 - No action taken - send the line  
 4 - Line to send has been changed - send new line  
 8 - Send the current line although it was not to be sent (valid only with operation code 5)  
 12 - Do not send the line due to be sent  
 16 - Cancel the submission of the entire job stream  
 20 - Send new lines instead of the current line. The lines and their number are indicated in parameters 7 and 8.

#### DSS0444 INVALID REQUEST CODE " " FOR D00YCLM

**Explanation:** This message indicates a logic error in the JOB/SCAN or DOCU/TEXT interface to CONTROL-M.

**Action:** Contact ASG Customer Support.

#### DSS0445 UNABLE TO ALLOCATE CONTROL-M DD " "

**Explanation:** The CONTROL-M interface requires the allocation of two files.

**Action:** These DD statements may be specified via JCL statements in a batch run, via TSO ALLOCATE commands in a CLIST, or via FILE ALLOCATIONS statements in the PARMLIB. This message is printed to the system summary report (JLSSR) if one or more of these DD's is not allocated. The



two DDs are: DAGLOBAL - Defines a library which contains global Auto Edit variables. PRTDBG - This DD is used for internal CONTROL-M debugging. It may be defined as DUMMY.

#### DSS0446 CONTROL-M EXIT 2 RETURNED ZERO ADD-ON COUNT

**Explanation:** The CONTROL-M EXIT 2 routine returned a return code of 20 (send new lines instead of original) but the count of the new lines to send was zero.

**Action:** This indicates a problem in the CONTROL-M EXIT 2 routine.

#### DSS0447 CONTROL-M CANCELLED THIS JOB - jobname

**Explanation:** The PROGRAM PRODUCT INTERFACE (PPI) Run Option indicated that CONTROL-M was active on the system and that the QUIT parameter was specified, this causes DOCUTEXT to terminate processing when CONTROL-M specifies that the current job should be flushed.

**Action:** Contact ASG Customer Support.

#### DSS0451 INTERNAL ERROR IN KEYWORD SEARCH

**Explanation:** Internal error in processing the mutually exclusive keyword table.

**Action:** Contact ASG Customer Support.

#### DSS0452 INTERNAL ERROR IN KEY REMATCH

**Explanation:** Internal error in processing the mutually exclusive keyword table.

**Action:** Contact ASG Customer Support.

#### DSS0500 MIXED RELEASES - xxxx and yyyy

**Explanation:** This message is produced when the JOB/SCAN or DOCU/TEXT supervisor has loaded a sub-routine that is not at the same release level. xxxx is the supervisory release level and yyyy is the subroutine release level. Subroutines may be loaded from the JLLD DD statement or ISPLLIB allocation. Make sure these libraries are allocated to load libraries that are the same release level as the JOBLIB/STEPLIB/LINKLIST libraries.

**Action:** If you are using LIBDEF's, check these allocations as well.

#### DSS0501 INVALID xxx >yyy< FOUND IN PARM FIELD

**Explanation:** When parsing for xxx, invalid character string yyy was found.

**Action:** Contact ASG Customer Support.

#### DSS0502 INVALID PARM FIELD

**Explanation:** The parm field on the EXEC statement of the batch processor is invalid.

**Action:** Most processing can be done with no parm field.

#### DSS0503 ROCESSOR MODULE xxx CANNOT BE LOADED

**Explanation:** The specified module cannot be dynamically loaded.

**Action:** Check allocations for ISPLLIB, JLLD, and STEPLIB. This is probably an installation error when specifying ISPLLIB or JLLD. Refer to the procedures in [Installation](#). NOTE: Concatenate ASG's

ISPLLIB ahead of existing datasets. If you choose to allocate DOCU/TEXT and JOB/SCAN libraries using ISPF LIBDEF Service, you must allocate a JLLOD DD statement. JLLOD DD statement points to the same library as your DOCU/TEXT ISPEXEC LIBDEF ISPLLIB statement.

#### DSS0504 NOT AUTHORIZED TO USE ONLINE XREF

Explanation: Your password is incorrect or you are not authorized to use ONLINE XREF.

Action: Contact ASG Customer Support.

#### DSS0505 NOT AUTHORIZED TO USE INFO/X

Explanation: An error message is issued when user is not authorized to use "INFO/X".

Action: Contact your INFO/X Product Administrator.

#### DSS0506 ALREADY RUNNING IN SPLIT SCREEN MODE - CANNOT START AGAIN

Explanation: The product may only be used in one split-screen ISPF session at a time.

Action: Contact ASG Customer Support.

#### DSS0507 INVALID RETURN xxx FROM INITIALIZATION

Explanation: There was at least one error in the contents of your PARMLIB. Processing cannot be continued.

Action: Use the Administrator function called "Test an Options Member" to scan these options members for errors.

#### DSS0508 NO PARM FIELD PASSED TO EDIT SUPERVISOR

Explanation: This is an internal error.

Action: Contact ASG Customer Support.

#### DSS0509 RETURN CODE xxxx PASSED FROM xxx

Explanation: This is an internal error. When running Reformatter, this error indicates JCL is syntactically incorrect.

Action: Run the JOB/SCAN process and correct the JCL prior to inputting it to Reformatter. If there is still a problem, contact ASG Customer Support.

#### DSS0510 CONTROL MEMBER xxx MISSING

Explanation: The PARMLIB member was not found in JLRCL. JLRCL is the DDNAME of the PARMLIB Library.

Action: Contact ASG Customer Support.

#### DSS0511 INVALID STATEMENT IN CONTROL MEMBER xxx

Explanation: The PARMLIB member has an erroneous statement.

Action: Use the Administrator function called "Test an Options Member" to scan the options members for errors.

#### DSS0512 xxx ACCESS DENIED. SEE ADMINISTRATOR

- Explanation: ACCESS CONTROL denies use of the indicated product. ACCESS CONTROL statement may be missing from the PARMLIB.
- Action: You may not own the module you wish to open. Your Product Administrator (on-site) has not included you in the access list. Your Product Administrator may have mis-keyed the access password. You may have recently changed your CPU without contacting ASG to obtain a new access password.

**DSS0513 CANNOT ALLOCATE SCREEN FOR BROWSE**

- Explanation: No JLSCRN in the FILE ALLOCATIONS.
- Action: Contact ASG Customer Support.

**DSS0514 CANNOT OPEN THE SELECTED JCL FILE**

- Explanation: This message is issued after any message DSS0701 - DSS0711.
- Action: Contact ASG Customer Support.

**DSS0515 CANNOT ALLOCATE INTERNAL READER - DDN JLINTRDR**

- Explanation: Verify the FILE ALLOCATIONS statement in the PARMLIB for ddname JLINTRDR has been coded correctly.
- Action: Contact ASG Customer Support.

**DSS0516 CANNOT INTERPRET PRODUCT CODE**

- Explanation: An edit macro may have been modified to pass an invalid parameter (product code) to an internal routine.
- Action: The acceptable codes are:

PRODUCT CODE	NAME OF JOB/SCAN EDIT MACRO
C	JCHG
E	JEM and JEMSUB
J	JSCAN and JSUB
M	JMSG
O	JSOPC
R	JREF
V	JVER

See also message DSS0631.

**DSS0517 PROBABLE INPUT JCL ERRORS; SUGGEST USING JOB/SCAN**

- Explanation: Your JCL may be syntactically incorrect.
- Action: If running JOB/SCAN does not resolve the problem, contact ASG Customer Support. If you are in an ISPF Edit session, JEM Edit Macro can be executed while still in Edit. JCL is validated, errors are flagged and corrections can be made. Use JMSG Edit Macro to receive expanded error messages.

#### DSS0518 UNRECOGNIZED PRODUCT IDENTIFIER

- Explanation: This message is produced when the product's ISPF supervisor is unable to determine which product's ISPF panels to display because of an incorrect product identifier.
- Action: Contact ASG Customer Support.

#### DSS0525 "product usage error message text"

- Explanation: An error occurred when reading or writing the JLLMRF Product Usage File. The error message text explains the specific error. Some messages are for ASG internal use.
- Action: If the error message is unclear, contact ASG Customer Support. Possible product usage error message texts are:

DSS0525 ATTEMPT TO STORE RECORD WITH DUPLICATE KEY  
 Attempted to store a record with a duplicate key

DSS0525 CANNOT CLOSE DDNAME=XXXXXXXX WHICH IS ALREADY  
 CLOSED Close OK, warning message issued

DSS0525 DDNAME=XXXXXXXX IS NOT ALLOCATED TO A VALID FILE  
 File structure is not valid for the Product Usage File

DSS0525 GENCB BLK=ACB FAILED Unable to create RPL for the  
 Product Usage File

DSS0525 INVALID PCF FUNCTION CODE.CHECK  
 PCFPARM.FUNCTIONCODE Requested function code not found

DSS0525 KEYED READ REQUEST FAILED Keyed read request has  
 failed

DSS0525 OPEN FAILED: DDNAME=XXXXXXXX IS NOT A VSAM DATA  
 SET Make sure the file allocated is a VSAM file

DSS0525 OPEN FAILED: ERROR IN THE VSAM VOLUME DATA SET  
 (VVDS) Make sure the file allocated to JLLMRF DDNAME is a VSAM  
 dataset

DSS0525 OPEN FAILED: MISSING DD STATEMENT FOR DDNAME-  
 E=XXXXXXXX Missing DD statement JLLMRF

DSS0525 OPEN OK, CHECK VSAM REASON CODE=NN Open OK,  
warning messages issued

DSS0525 POINT REQUEST FILED Point request has failed

DSS0525 SEQUENTIAL READ REQUEST FAILED Sequential read  
request has failed

DSS0525 SEQUENTIAL WRITE REQUEST FAILED Sequential write  
request has failed

DSS0525 UNEXPECTED D00YPCF RETURN CODE This is an internal  
error in module D00YPCF

DSS0525 VSAM CLOSE ERROR: RETURN CODE=NN, REASON CODE=  
E=NN There was a VSAM CLOSE error encountered

DSS0525 VSAM PUT ERROR; RETURN CODE=NN, REASON CODE=NN  
There was a VSAM PUT error encountered

#### DSS0533 STDS/SEL/PGM/RPT EXIT "XXXXXXXX" INCOMPATIBLE WITH CXE

Explanation: The STANDARDS, SELECTION, PROGRAM NAME or REPORT EXIT coded is not compatible with the COBOL EXIT ENVIRONMENT (CXE) coded in the JOB/SCAN, DOCU/TEXT or INFO/X Run Options. Processing is terminated to avoid performance or other problems.

Action: Correct the CXE or COBOL EXIT ENVIRONMENT (CXE) to match the Cobol level used to compile your exit or exits. The Default is none (no Cobol exits).

#### DSS0550 INPUT FILE xxx ALLOCATED TO SYSOUT

Explanation: A FILE ALLOCATIONS request has mistakenly allocated an input file to sysout.

Action: Verify the FILE ALLOCATIONS statements coded in the PARMLIB.

#### DSS0551 VALUE FOR xxx REQUIRED AND MISSING

Explanation: A required parameter was missing from the FILE ALLOCATIONS statement.

Action: Verify the FILE ALLOCATIONS statements in the PARMLIB. Refer to [File Allocations Group](#) for valid parameters when coding FILE ALLOCATIONS statements.

#### DSS0552 REQUEST CODE xxx to DY2 IS INVALID FOR xxx

Explanation: Internal error. The stated request code is invalid; therefore, the specified ddname will not be processed.

Action: Contact ASG Customer Support.

#### DSS0553 DISPOSITION xxx INVALID FOR INPUT FILE

Explanation: The stated DISP code is invalid for an input file.

Action: Verify the DISP parameters coded for FILE ALLOCATIONS statements in the PARMLIB.

#### DSS0554 NO SYSOUT CLASS RESOLVED FOR xxx VALUE xxx

- Explanation: Other values specified in the FILE ALLOCATIONS statement indicate this is a SYSOUT request; however, no class was specified or could otherwise be determined.
- Action: Verify the FILE ALLOCATIONS statements coded in the PARMLIB.

**DSS0555 NO DIRECTORY SPACE FOR NEW PO DDNAME xxx**

- Explanation: The DSORG specified this as a newly created PO dataset, but no directory allocation was specified.
- Action: Verify the FILE ALLOCATIONS statements coded in the PARMLIB.

**DSS0557 NO UNIT FOR DDNAME xxx WITH DISP xxx**

- Explanation: A UNIT specification is required for a new dataset. Verify the FILE ALLOCATIONS statements coded in the PARMLIB.
- Action: Refer to [File Allocations Group](#) for information regarding FILE ALLOCATIONS statements and their parameters.

**DSS0558 DISP 2 FOR DDNAME xxx HAS INVALID VALUE xxx**

- Explanation: The disposition at deallocation was improperly specified. Verify the FILE ALLOCATIONS parameters coded in the PARMLIB.
- Action: Refer to [File Allocations Group](#) for information regarding FILE ALLOCATIONS statements and their parameters.

**DSS0559 NO BLOCKSIZE FOR BLOCK ALLOCATION DDNAME xxx**

- Explanation: An allocation was requested in blocks, but no blocksize was included.
- Action: Verify the FILE ALLOCATIONS parameters coded in the PARMLIB. Refer to [File Allocations Group](#) for information regarding FILE ALLOCATIONS statements and their parameters.

**DSS0570 REQUIRED DD dddddddd NOT FOUND**

- Explanation: The DDNAME 'ddddddd', specified in the message, which was required to run the OPC/A flow-chart data extractor, was not allocated for this run.
- Action: Dynamic allocation may be attempted, as indicated in a subsequent message.

**DSS0571 DYNAMIC ALLOCATION WILL BE ATTEMPTED**

- Explanation: Follows message DSS0570 for those DDNAMES that can be allocated to a 'default' category (SYSOUT, for example). Note that the dynamic allocation, even if successful, may not match the user's intent, if the DDNAME had been present.
- Action: It is always safer to include the specific allocation via a DD statement.

**DSS0572 REQUIRED DD dddddddd NOT FOUND**

- Explanation: The DDNAME 'ddddddd', specified in the message, which was required to run the OPC/A flow-chart data extractor, was not allocated for this run.

Action: Dynamic allocation cannot be attempted, because there is no adequate 'default' that could be presumed for this DD.

#### DSS0573 ALLOCATE MANUALLY AND RE-TRY

Explanation: Follows message DSS0572 for those DDNAMES that cannot be allocated to a 'default' category.

Action: Include a DD statement for the specified file and re-submit the job.

#### DSS0574 DYNALLOC OF dddddddd FAILED

Explanation: The attempt to dynamically allocate the DDNAME 'ddddddd' to a default category has failed.

Action: Include a DD statement for the given file and re-submit the job. Possible causes include the use of non-standard UNIT names at a site (the default is 'SYSDA').

#### DSS0575 DYNALLOC OF dddddddd SUCCESSFUL

Explanation: The attempt to dynamically allocate the DDNAME 'ddddddd' to a default category has succeeded.

Action: Note, however, that the category selected might not agree with the user's intent, and that it is always safer to include an explicit DD statement allocating the indicated file.

#### DSS0576 EXCESSIVE PARM LENGTH-TRUNCATED TO 4

Explanation: Only a 4-character OPC/A subsystem ID may be passed as a PARM to the OPC/A flowchart data extractor.

Action: The first 4 characters will be used as the subsystem ID; the rest will be ignored.

#### DSS0577 xxxYCOM xxxxxxxx REQUEST, RC=rr

Explanation: A generic message used to report unexpected results from the CSYYCOM or EQQYCOM IBM-supplied OPC/A (and OPC/ESA) interface modules. Issued by the OPC/A flowchart data extractor when unanticipated results occur.

Action: Report the specifics of the message for investigation.

#### DSS0578 OPEN FAILED FOR DD dddddddd (INPUT)

Explanation: The attempt to OPEN the DDNAME 'ddddddd' for input has failed.

Action: Usually caused by some type of allocation problem; check the JCL, and, if unable to determine the cause, report to ASG Customer Support.

#### DSS0579 NO RECS FOR aaaaaaaaaaaaaaaaa (OR NOT AUTH)

Explanation: In the OPC/A flowchart data extractor program, no records were found for the application 'aaaaaaaaaaaaaaaa'.

Action: This could also be caused by lack of authorization for the given user to access the OPC/A resource. Check the spelling of the application ID and correct it if necessary.

#### DSS0580 OPEN FAILED FOR DD dddddddd (OUTPUT)

Explanation: The attempt to OPEN the DDNAME 'ddddddd' for output has failed.

Action: Usually caused by some type of allocation problem; check the JCL, and, if unable to determine the cause, report to ASG Customer Support.

#### DSS0581 NO GRAPH RECORDS PULLED.

Explanation: In the OPC/A database, no dependency data was found which matched the search criteria specified in the control cards for the flow-chart data extractor.

Action: Check the control cards for possible typographical errors.

#### DSS0582 INVALID OPC/ESA INTERFACE INPUT CARD

Explanation: The input file contains data that was not recognized. The input has been ignored.

Action: Check to ensure that the input file contains only the allowed keywords spelled properly and beginning in the proper columns.

#### DSS0583 DUPLICATE OPC/ESA INTERFACE INPUT CARD

Explanation: The input file contains duplicate data and has been ignored.

Action: Look at the input data and eliminate duplicate keyword parameters.

#### DSS0584 OPC/ESA INTERFACE WS RECORD MANDATORY

Explanation: The input file is required to contain the workstation parameter which indicates the workstation ids to use in selecting jobs from OPC/ESA.

Action: Make sure the input data contains the WS parameter.

#### DSS0585 OPC/ESA INTERFACE DATE/TIME MISSING

Explanation: The input file contains an invalid combination of date and time parameters.

Action: The date and time parameters are optional but both are required if one is specified.

#### DSS0586 OPC/ESA INTERFACE DATE/TIME BAD

Explanation: The input file contains an invalid combination of date and time parameters.

Action: The date and time parameters are optional but when used are restricted to using the DATE and TIME parameters, or the SDATE, STIME, EDATE and ETIME parameters to limit the selection of jobs for a time period.

#### DSS0587 OPC/ESA INTERFACE AID1 > AID2

Explanation: The input file contains an invalid combination of AID1 and AID2 parameters.

Action: When both AID1 and AID2 are specified, they will be used to select a range of application ids, and AID2 must be higher than AID1.

#### DSS0588 OPC/ESA INTERFACE NO OPS QUALIFY

Explanation: There were no selections returned from OPC/ESA using the criteria given in the input file.

Action: Check the input specifications on the input data.

#### DSS0589 OPC/ESA INTERFACE SIMTIME VALUE INVALID



Explanation: The SIMTIME value provided contains an invalid date or time parameter and must conform to the CCYYMMDDHHMM format.

Action: Although the field is optional, the values passed in the date and time values must be valid year, month, day (for those months), hour and minute values. (e.g. Feb 30th, hour 25, year 1895, minute 61 invalid). Note also that OPC limits year value from 1984 to 2071.

#### DSS0590 OPC/ESA INTERFACE - SCAN SUBMIT PARAMETER INVALID

Explanation: The SCAN= parameter provided does not contain a correct value.

Action: VALID values are ALL in which case all jobs (regardless of their OPC SUBMIT:Y or SUBMIT:N values) are scanned or ONLYIFSUBMITY in which case only jobs coded with SUBMIT:Y on their applications automatic JOB details will be scanned. ONLYIFSUBMITY would be coded if the jobs in this application may come from outside the OPC/ESA EQQJBLIB concatenation. ONLYIFSUBMITY means that ANY JOB with SUBMIT=N will NOT be validated as in the library. When a JOB is scanned, a DSS0588E will be issued if the job is not found within the library. Correct the Parm and resubmit the JCL stream.

#### DSS0595 IA DATE SPECIFIED IS PAST THE END OF THE CURRENT PLAN.

Explanation: SOME VARIABLES MAY NOT RESOLVE AS EXPECTED.

Action: The Input Arrival date specified to the JSOPC edit macro was past the end of the current plan. SIMTIME will be used to specify the IA date to OPC. The date/time variables based on the occurrence will only resolve to the end of the current plan. This message is inserted into JCL being validated as a JCL comment.

#### DSS0600 DSN = dsname

Explanation: The Data Set Name referred to in the previous message.

Action: Contact ASG Customer Support.

#### DSS0601 OPEN ERROR FOR xxx CODE xxx

Explanation: A member oriented library (PDS, PANVALET, or LIBRARIAN) whose DDNAME is specified, has failed to open.

Action: The Library Type field may be incorrectly specified. The code indicates the type of failure. Consult your Product Administrator for assistance.

#### DSS0602 MEMBER xxx IS EMPTY.

Explanation: The member specified contains no records.

Action: Contact ASG Customer Support.

#### DSS0603 INVALID SOURCE CODE xxx RECEIVED, REQUEST xxx

Explanation: Internal error. The Library Type field specified is incorrect.

Action: Contact ASG Customer Support.

#### DSS0604 INVALID REQUEST CODE xxx RECEIVED, REQUEST xxx

- Explanation: Internal error. The request code from an application processor to the member read routine was incorrect.
- Action: Contact ASG Customer Support.

#### DSS0605 INTERFACE MODULE xxx NOT FOUND

- Explanation: The PANVALET or LIBRARIAN interface module named could not be loaded from ISPLLIB, JLLOD, or STEPLIB. This could be an installation problem.
- Action: Verify the correct external processor link was established. Refer to [Library Management Interfaces](#), specifically the installation of PANVALET or LIBRARIAN Optional Interfaces.

#### DSS0606 xxx RETURNED INVALID CODE xxx FROM CALL

- Explanation: An unexpected or invalid code was returned from a library interface processor (PANVALET) to member read. This could be an installation problem.
- Action: Verify that the correct external processor link was completed successfully. See [Library Management Interfaces](#) documentation.

#### DSS0607 INVALID FORMATTING OPTIONS: xxx

- Explanation: Internal error. The specification of return formatting to the member read routine was incorrect.
- Action: Call ASG Customer Support.

#### DSS0608 ISPF PACKED MEMBER membername INVALID FOR ddname

- Explanation: Member named 'membername' was read and found to possibly be in the ISPF EDIT 'PACK ON' format. The dataset associated with the 'ddname' is named in the following DSS0600 message.
- Action: Edit the data set member and enter the EDIT profile command 'PACK OFF'. Save the member and resubmit the JOB.

#### DSS0609 ZERO ADDRESSES TO PAM INTERFACE

- Explanation: The PANVALET interface module D00YPAM has not been properly linked. The entry point addresses to the service routines are zero. This is an installation problem.
- Action: Refer to [Library Management Interfaces](#) for more information on PANVALET.

#### DSS0610 ZERO ADDRESSES TO LIB INTERFACE

- Explanation: The LIBRARIAN interface module D00YLIB3 has not been properly linked. The entry point addresses to the service routines are zero. This is an installation problem.

#### DSS0611 DDNAME, OPERATION ATTEMPTED, ERROR DESCRIPTION, TRACK ADDRESS, ACCESS METHOD

- Explanation: An uncorrectable I/O error has occurred while processing the DD statement named "ddname". An end-of-file is forced for this dataset and processing continues.
- Action: Contact ASG Customer Support.

#### DSS0612 DIRECTORY OPEN FAILED FOR xxxxxxxx (SUBSYS)

Explanation: DDNAME xxxxxxxx has been allocated with the "SUBSYS=" parameter and a search for a member was attempted using a "mask" or a "range" of members. Mask or range cannot be used if the dataset has been allocated using the SUBSYS parameter and the selection type was either omitted or specified as PDS.

Action: See [Library Management Interfaces](#) for information on selecting members residing in LIBRARIAN or PANVALET datasets.

#### DSS0613 SECURITY VIOLATION FOR MEMBER xxxxxxxxxx

Explanation: Access to member xxxxxxxxxx has been denied by the security system.

Action: The member is treated as though it does not exist.

#### DSS0614 PANVALET ERROR xxxxx PROCESSING MEMBER membername

Explanation: PANVALET has issued an error message while processing the following dataset (see DSS0615).

Action: The message is documented in the PANVALET MVS System Management Manual.

#### DSS0615 OF DSN dsname

Explanation: This is the second half of the previous DSS0614 message and shows the PANVALET dataset name.

Action: Contact ASG Customer Support.

#### DSS0616 PANVALET ERROR xxxxx READING ddname DIRECTORY

Explanation: An error has occurred attempting to read the PANVALET directory of the named ddname.

Action: The PANVALET error code is documented in the PANVALET MVS System Management Manual.

#### DSS0617 ERROR UPDATING RANGE FOR PANVALET MEMBER membername

Explanation: An error has occurred attempting to update the current PANVALET member name within a range of member names.

Action: Contact ASG Customer Support.

#### DSS0631 PRODUCT CODE xxx IS INVALID

Explanation: Internal error. The specified product code was incorrectly passed to the RLSE subroutine.

Action: Check that the Parm on the PROC statement was correct. Contact ASG Customer Support.

#### DSS0651 CATALOG FOR xxx NOT MOUNTED

Explanation: The return code from accessing a CVOL indicated the volume was not currently mounted.

Action: Consult your systems programmer.

#### DSS0652 I/O ERROR ON CATALOG FOR xxx

Explanation: The return code from accessing a CVOL indicated an I/O error was encountered. Possible CVOL damage.

Action: Consult your systems programmer.

**DSS0653 DATA SET xxx NOT CATALOGED**

Explanation: The return code from accessing a CVOL indicated the dataset specified was not found.

Action: Contact ASG Customer Support.

**DSS0654 CAT READ FOR xxx IS SECURITY VIOLATION**

Explanation: The return code from accessing a CVOL indicated insufficient access authority to read a catalog.

Action: Contact ASG Customer Support.

**DSS0655 DSN FORMAT FOR xxx INVALID IN CATALOG**

Explanation: The format of the dataset is invalid.

Action: It cannot be contained in a CVOL.

**DSS0656 VOL-SER TABLE OVERFLOW FOR xxx**

Explanation: The named dataset resides on more volumes than the system can table.

Action: The list of volumes is truncated for only this dataset. Processing continues.

**DSS0660 INTERNAL LIBRARY STACK LIMIT EXCEEDED FOR xxx**

Explanation: The number of DD cards in a "lib-type" (ddname, 'PROCLIB', 'IMS' or 'PECNTL') concatenation has exceeded 64 (DOCU/TEXT - JOB/SCAN limit).

Action: Reduce the cards.

**DSS0684 NO RECORDS WERE PROCESSED BY JCLSCAN PROGRAM J00YESPX**

Explanation: No JCL records were encountered or processed by J00YESPX. J00YESPX is called by ESP as the result of a JCLSCAN sub-parameter on a ESP SIMULATE command issued by JOB/SCAN.

Action: To obtain additional information use the DEBUG sub parameter on the JSESP edit macro command. Example: JSESP DEBUG

**DSS0701 OPEN ERROR FOR xxx for xxxING**

Explanation: The sequential file accessor could not open the specified ddname for either reading or writing, as shown.

Action: Verify that the dataset exists, it is cataloged, and that you have access authority to it.

**DSS0702 NO RECORD ADDRESS FOR xxx OF xxx**

Explanation: Internal error. No record address was provided to the sequential file accessor.

Action: Contact ASG Customer Support.

**DSS0703 FOR DDN xxx STATUS xxx READING AN OUTPUT FILE**

Explanation: Internal error.

Action: Contact ASG Customer Support.

**DSS0704 FOR DDN xxx STATUS xxx FILE HAS FAILED OPEN**

Explanation: Internal error.  
Action: Contact ASG Customer Support.

**DSS0705 FOR DDN xxx STATUS xxx WRITING AN INPUT FILE**

Explanation: Internal error.  
Action: Contact ASG Customer Support.

**DSS0706 FOR DDN xxx STATUS xxx FILE IS ALREADY OPEN**

Explanation: Internal error.  
Action: Contact ASG Customer Support.

**DSS0707 FOR DDN xxx STATUS xxx FILE IS ALREADY CLOSED**

Explanation: Internal error.  
Action: Contact ASG Customer Support.

**DSS0708 FOR DDN xxx STATUS xxx UNKNOWN REQUEST-STATUS**

Explanation: Internal error.  
Action: Contact ASG Customer Support.

**DSS0709 NO TABLE POINTER FOR xxx OF xxx**

Explanation: Internal error. For a sequential file carried as an internal table, there is no pointer to the table.  
Action: Contact ASG Customer Support.

**DSS0710 FOR DDN xxx STATUS xxx - FILE ACCESS FAILED**

Explanation: Internal error.  
Action: Contact ASG Customer Support.

**DSS0711 DDNAME, OPERATION ATTEMPTED, ERROR DESCRIPTION, TRACK ADDRESS, ACCESS METHOD**

Explanation: An uncorrectable I/O error has occurred while processing the DD statement named "ddname".  
An end-of-file is forced for this dataset and processing continues.  
Action: Contact ASG Customer Support.

**DSS0715 DDNAME nnnnnnnn CONTAINS ISPF PACKED DATA**

Explanation: The dataset described by the specified DD name was read and found to possibly be in the ISPF 'PACK ON' format.  
Action: Edit the dataset and enter the ISPF EDIT profile command 'PACK OFF'. Save the dataset and resubmit the JOB.

**DSS0720 RECORD FOR JOBNAME XXXXXXXX CONTAINS INVALID DATE FIELD.**

Explanation: DOCU/TEXT has found a field in the JLLTM file (VSAM XREF FILE) which does not contain a valid date. Although processing continues, the date processing for this jobname is skipped to avoid abends.

Action: Contact ASG Customer Support.

#### DSS0721 SELECTION ENTRY HAS INVALID FORMAT:

Explanation: While ONLINE XREF QUERY (D50XBQX) was reading the control card file, an entry was found which did not correspond to the valid options for an OLX query. Valid are JOB, PROC, SYSOUT, PROGRAM or FILE.

Action: See the OLXQIN or DOCU/TEXT User Information for valid selection options.

#### DSS0722 "\*\*\*\*" IS AN INVALID MASK

Explanation: DOCU/TEXT ONLINE XREF QUERY (D50XBQX) has found a control statement mask with all asterisks (not valid).

Action: See SAMPLIB OLXQIN member or DOCU/TEXT User Information under "ONLINE XREF: Querying in Batch Mode" for allowable formats.

#### DSS0723 "\*\*\*\*" NOT ALLOWED IN MEMBER

Explanation: DOCU/TEXT ONLINE XREF QUERY (D50XBQX) has found a control statement with a member name containing multiple asterisks.

Action: Refer to OLXQIN member of SAMPLIB or DOCU/TEXT User Information under "ONLINE XREF: Querying in Batch Mode" for allowable format. For example W FILE =DSSI.\$DTSTG.SAMPLIB(\*\*) (not allowed).

#### DSS0724 "\*\*\*\*" NOT VALID WITHIN A MASK FIELD.

Explanation: DOCU/TEXT ONLINE XREF QUERY (D50XBQX) has found a control statement with a MASK which contains an invalid format.

Action: Refer to OLXQIN member of SAMPLIB or DOCU/TEXT User Information under "ONLINE XREF: Querying in Batch Mode" for allowable format. Double asterisks are not permitted within a name (e.g. IEF\*\*14).

#### DSS0725 MASK EXCEEDED MAXIMUM LENGTH

Explanation: DOCU/TEXT ONLINE XREF QUERY (D50XBQX) has found a control statement with a mask field which contains an illogical mask.

Action: Refer to member of SAMPLIB OLXQIN for samples and resubmit the JCL. An example might be D FILE =DSS\*M.XREF%%%%%%%%\*.FILE.INDEX

#### DSS0726 JLLTM OPEN ERROR - VSAM CD= XX.

Explanation: DOCU/TEXT ONLINE XREF QUERY (D50XBQX) issued an open to the JLLTM (OLX/SDF) dataset. The open failed with this status key value.

Action: See additional DSS073\* message for more information. NOTE: These values are listed in the COBOL documentation under "VSAM File Processing".

#### DSS0727 VSAM READ ERROR - VSAM CD = XX.

Explanation: DOCU/TEXT ONLINE XREF QUERY (D50XBQX) READ to the JLLTM VSAM FILE failed with this status key value.

Action: These values are documented under "VSAM File Processing" in the Cobol documentation.

#### DSS0729 WHERE USED "W" NOT VALID FOR JOB.

Explanation: DOCU/TEXT ONLINE XREF QUERY (D50XBQX) found a control statement with a "W" for WHERE USED.

Action: This option is not available for a JOB item. Only "D" is valid (for directory display).

#### DSS0730 CONTROL STATEMENT NOT VALID.

Explanation: DOCU/TEXT ONLINE XREF QUERY (D50XBQX) found a JOB item request with an unknown option.

Action: "D" for directory display may be used and the JCL resubmitted.

#### DSS0731 NO REPORT FORMAT CHOSEN.

Explanation: DOCU/TEXT ONLINE XREF QUERY (D50XBQX) JCL submitted failed to request a known format for JOBS. Note that Directory display is only valid format for JOBS = . (Use "D" and resubmit).

Action: Contact ASG Customer Support.

#### DSS0732 X IS NOT A VALID REPORT FORMAT

Explanation: DOCU/TEXT ONLINE XREF QUERY (D50XBQX) found a control statement that does not contain a valid report format (either W or D).

Action: Correct and resubmit. See ONLINE XREF: Querying in Batch Mode in DOCU/TEXT User Information.

#### DSS0733 UNABLE TO OPEN - UNCORRECTABLE IO ERROR.

Explanation: D50XBQX (Batch ONLINE XREF QUERY) failed with a permanent I/O error when trying to open the dataset named in your JLLTM DD card. VSAM Status Code = 30.

Action: Contact the person supporting your SDF/OLX file for assistance.

#### DSS0734 UNABLE TO OPEN - AUTHORIZATION FAILURE

Explanation: D50XBQX (Batch ONLINE XREF QUERY) tried to open the JLLTM file and got a 91 VSAM status key value.

Action: This is because your JLLTM DD card names a file for which you do not have RACF authority.

#### DSS0735 OPEN ERROR - LOGIC ERROR OR ALREADY OPEN

- Explanation: DOCU/TEXT ONLINE XREF QUERY (D50XBQX) was unable to open the OLX VSAM dataset due to a possible logic error in the program or maybe due to another application which has the file open.
- Action: Contact ASG Customer Support.

#### DSS0736 UNABLE TO OPEN - NOT ENOUGH VIRTUAL STORAGE

- Explanation: DOCU/TEXT ONLINE XREF QUERY (D50XBQX) WAS UNABLE TO OPEN THE OLX VSAM dataset because there was insufficient virtual storage available. VSAM status key value = 93.
- Action: Increase the region size on your JOB statement and resubmit.

#### DSS0737 UNABLE TO OPEN - FILE NOT NEW SDF-XREF FORMAT.

- Explanation: DOCU/TEXT ONLINE XREF QUERY (D50XBQX) WAS UNABLE TO OPEN THE OLX VSAM dataset because the dataset does not match the required format for it. The JLLTM file should be a VSAM KSDS . Check the JLLTM dd card. VSAM status key value = 95.
- Action: See Create the OLX repository in DOCU/TEXT User Information or the person who installed the DOCU/TEXT product.

#### DSS0738 UNABLE TO OPEN - NO JLLTM FILE ALLOCATION.

- Explanation: DOCU/TEXT ONLINE XREF QUERY (D50XBQX) WAS UNABLE TO OPEN THE OLX VSAM dataset because the DD card was missing or is incorrect or the file was unable to be allocated.
- Action: Correct and resubmit the job. VSAM status key value = 96.

#### DSS0739 NO MATCH FOUND IN SDF/OLX FILE

- Explanation: DOCU/TEXT ONLINE XREF QUERY (D50XBQX) read the SDF/OLX file but was not able to find a match using the control statement mask provided.
- Action: Check your control statements or perhaps the JLLTM DD card is pointing to the wrong database.

#### DSS0740 NO VALID INPUT FOUND TO D50XBQX

- Explanation: DOCU/TEXT ONLINE XREF QUERY (D50XBQX) read the SDF/OLX SYSIN file provided but EOF was received prior to finding a valid control statement.
- Action: Check for other messages and compare your input to valid samples listed in the OLXQIN member of Samplib or in DOCU/TEXT User Information under "ONLINE XREF: Querying in Batch Mode".

#### DSS0750 UNKNOWN FUNCTION CODE RECEIVED - XX

- Explanation: Internal error. The Product Usage File manager received an unknown function directive.
- Action: Contact your JOB/SCAN administrator.

#### DSS0751 USER xx UNKNOWN TO PRODUCT USAGE FEATURE



Explanation: Internal error. While attempting to deactivate or reactivate a user, no record of the selected user could be found.

Action: Contact your JOB/SCAN administrator.

#### DSS0752 PRODUCT USAGE FILE IS NOT ALLOCATED

Explanation: The Product Usage dataset is not allocated to the DD name JLLPUF.

Action: JLLPUF can be specified in options member JSOPT02.

#### DSS0753 OLD PRODUCT USAGE FILE IS CORRUPT.

Explanation: While converting an old style Product Usage file into the new format, a duplicate record was found.

Action: Contact your JOB/SCAN administrator for problem resolution.

#### DSS0754 MISSING OLD PRODUCT USAGE FILE. DD=JLLMRF

Explanation: When converting an old style Product Usage file into the new format, the DD name JLLMRF was not allocated.

Action: Ensure JLLMRF is allocated to the old style usage file before attempting the conversion.

#### DSS0755 MISSING NEW USAGE FILE. DD=JLLPUF

Explanation: When converting an old style Product Usage file into the new format, the DD name JLLPUF was not allocated.

Action: Before converting usage files, JLLPUF must be allocated to a new (and empty) usage file. See [Creating and Maintaining a Product Usage File](#) documentation for information on creating Product Usage files.

#### DSS0756 UNRECOVERABLE I/O ERROR ACCESSING JLLPUF

Explanation: A serious I/O error was encountered while attempting to read from or write to the Product Usage File, JLLPUF.

Action: Make sure JLLPUF is pointing to a valid file as documented in [System Administrator Tasks](#).

#### DSS0757 JLLPUF IS ALLOCATED TO A PRE JOB/SCAN RELEASE 6.2.6A USAGE FILE

Explanation: The Product Usage file allocated to the DD name JLLPUF was created by a version of JOB/SCAN prior to release 6.2.6A.

Action: As of 6.2.6A a new Product Usage file must be created and allocated to JLLPUF before the Product Usage feature can be enabled.

#### DSS0758 JLLPUDLT NOT ALLOCATED

Explanation: The sequential dataset used for formatting Product Usage data is not allocated.

Action: Contact ASG Customer Support.

#### DSS0759 CANNOT LOCATE SITE RECORD

Explanation: The usage file download utility, D00YPUDN, determined the Product Usage file was empty.

Action: JOB/CAN has not been run since the Usage file was created.

#### DSS0760 CANNOT LOCATE PERIOD RECORD

Explanation: The usage file download utility, D00YPUDN, determined the Product Usage file was corrupt.

Action: Contact your Product Administrator for assistance.

#### DSS0761 UNABLE TO INITIALIZE ENVIRONMENT

Explanation: The usage file download utility, D00YPUDN, was not able to establish an environment for running.

Action: Make sure your ASG load library is accessible to the D00YPUDN job.

#### DSS0762 UNDECIPHERABLE PERIOD NUMER(S) FOUND

Explanation: Incorrect parameters were passed to the Product Usage file utility D00YPUDN.

Action: Correct the parameters and re-run the job.

#### DSS0763 PRODUCT USAGE FILE IS NOT ALLOCATED

Explanation: When running the Product File download utility, the Product Usage file was not allocated to the DD name JLLPUF.

Action: Contact ASG Customer Support.

#### DSS0764 PERIOD nn NOT FOUND IN USAGE FILE

Explanation: One or more period numbers passed to the Product Usage File download utility were not found in the Product Usage File.

Action: Processing continues if more period numbers were given after this one.

#### DSS0765 USAGE REPORTING RECORD NOT WRITTEN

Explanation: If an error is encountered processing a Product Usage File, this message will follow one or more detailed messages explaining the nature of the error.

Action: Contact ASG Customer Support.

#### DSS0770 INVALID EDIT REQUIREMENT IDCAMS

Explanation: Internal error. An invalid value was found in field KYTFTYPE of macro D00RKEYD. Message is issued by program D00YCAMS.

Action: Contact ASG Customer Support.

#### DSS0777 IDCAMS VSAM ASSOCIATIONS TABLE ENTRY IS FULL

Explanation: This is an internal processing error.

Action: Contact ASG Customer Support.

#### DSS0801 STANDARDS PROGRAM xxxxxxxx IS NOT ACCESSIBLE

Explanation: The specified standards exit could not be located in ISPLLIB, JLLD, and STEPLIB.

Action: Check spelling of the exit name and the allocations of the above mentioned ddnames. Verify that the exit was linked into one of the load libraries contained in the concatenation lists of the previous ddnames. These ddnames may be found in your LOGON PROC, LOGON CLIST, DOCU/TEXT or JOB/SCAN Invocation CLIST, and/or the JOB/SCAN Edit Macros if you are running online. When running in batch, these allocations are done in your batch JCL, DOCU/TEXT PROC, or JOB/SCAN PROC.

#### DSS0802 NO FIRST CALL TO STANDARDS PROGRAM xxxxxxxx

Explanation: Internal error. The initial call to the standards program was not detected by the standards exit interface processor.

Action: Contact ASG Customer Support.

#### DSS0803 SECURITY EXIT xxx IS NOT ACCESSIBLE

Explanation: The specified exit could not be located in ISPLLIB, JLLOD, and STEPLIB. Check spelling of the exit name and the allocations of the above mentioned ddnames.

Action: Verify that the exit was linked into one of the LOAD libraries contained in the concatenation lists of the previous ddnames. These ddnames may be found in your LOGON PROC, LOGON CLIST, DOCU/TEXT or JOB/SCAN Invocation CLIST, and/or the JOB/SCAN Edit Macros if you are running online. When running in batch, these allocations are done in your batch JCL, DOCU/TEXT PROC, or JOB/SCAN PROC.

#### DSS0804 USER EXIT xxxxxxxx IS NOT ACCESSIBLE

Explanation: The REPORT exit, PROGRAM NAME exit, or other user exit named "xxxxxxx" could not be located.

Action: Check the spelling of the exit name and the allocations of ddnames JLLOD, ISPLLIB, and STEPLIB. Verify that the exit was linked into one of the load libraries contained in the concatenation lists of these ddnames or the LINKLST datasets or the link pack area (LPA). These ddnames may be found in your LOGON PROC, LOGON CLIST, DOCU/TEXT or JOB/SCAN Invocation CLIST, and/or the JOB/SCAN Edit Macros if you are running online. When running in batch, these allocations are done in your batch JCL, DOCU/TEXT PROC, or JOB/SCAN PROC.

#### DSS0806 ABEND xxxx IN USER EXIT yyyyyyyy - EP=zzzzzzzz

Explanation: The user exit named yyyyyyyy has abnormally completed with a completion code of xxxx. The entry point of the module is zzzzzzzz.

Action: Contact ASG Customer Support.

#### DSS0810 JSTABLE failed; Calling argument must be a 2 char table name

Explanation: JSTABLE was invoked with an invalid calling argument. There can be only one argument passed to JSTABLE and that argument must be the 2-character name of the table to be loaded.

Action: Contact ASG Customer Support.

#### DSS0811 JSTABLE failed; Required input parameter not specified

Explanation: JSTABLE requires one input parameter, the name of the table to be retrieved.

Action: Contact ASG Customer Support.

#### DSS0812 JSGET failed; Unknown field name passed as input

Explanation: JSGET was called to retrieve the value of an unknown field.

Action: Verify the spelling of the field name passed as an input parameter to JSGET.

#### DSS0813 JSMSG failed: Maximum of 50 messages per statement exceeded

Explanation: You can issue a maximum of 50 messages per JCL statement with JSMSG. If this limit is exceeded, an error is generated.

Action: You can monitor the number of messages currently issued by examining the return value from JSMSG. JSMSG returns the number of messages issued for the current JCL statement.

#### DSS0814 JSMSG failed; Single calling argument missing or invalid

Explanation: A single calling argument is passed to JSMSG that specifies the operation to be performed. The valid arguments are "ADD", "FIND", "SUPPRESS", and "CHANGE".

Action: Make sure your call of JSMSG includes one of these operation codes.

#### DSS0815 JSMSG failed; Fetch of REXX variables return code = xxxxxxxx

Explanation: JSMSG requires that certain REXX variables be initialized for each call type (e.g. "ADD", "CHANGE", etc.) This error is generated when JSMSG detects a failure retrieving the values of the required REXX variables. IRXEXCOM is the routine used to fetch REXX variable values. The likely IRXEXCOM return codes are: 1 - Variable not found (required REXX variable not initialized) 4 - Value truncated (the value is too long) check the maximum variable lengths for the JSMSG variables DSS0816 JSMSG failed; Invalid character passed for message level

Action: Verify MSG\_LEVEL value. MSG\_LEVEL is a 1 character variable that controls message output. Verify that you are initializing MSG\_LEVEL to one of the valid values.

#### DSS0817 JSMSG failed; Message severity is not an integer 0 - 99

Explanation: Message severity (REXX variable MSG\_SEVERITY) is a 1- or 2-digit positive integer.

Action: Check your initialization of MSG\_SEVERITY before the failing JSMSG call.

#### DSS0820 FATAL ERROR: SEQ seq1 FOR type name

Explanation: A fatal write error occurred when attempting to read the JLLTM file. The sequence number received for the specified record name and type was not contiguous.

Action: Restore the JLLTM file from the backup tape and re-run.

#### DSS0821 FATAL ERROR VSAM CODE(90). RESTORE FILE FROM BACKUP.

Explanation: A fatal write error occurred when attempting to update JLLTM. The SDF/XREF file probably ran out of space.

Action: Reallocate the file with more primary space, restore the file from the backup tape and re-run this XREF update run.

#### DSS0822 DUPLICATE JOB NAME ON INPUT- name

Explanation: Jobs with duplicate names were detected in the XREF Update input stream. A warning message is issued for the job, but XREF Update processing will continue.

Action: To eliminate the warning message, modify your JCL members to contain unique job names.

#### DSS0823 ERROR - rc ON ENQ FOR JLLTM

Explanation: Error on ENQ for JLLTM. File is in use.

Action: Try again at a later time.

#### DSS0824 INVALID DELETE COMMAND - word

Explanation: The XREF Delete command, as specified in PARMLIB, was invalid. Correct specification is "XRDELETE=".

Action: Refer to Maintenance: How to Delete a Job in DOCU/TEXT User Information for further information about deleting items from the SDF/XREF File. CONTROL TABLES is the Group Level statement XRDELETE=membrname is the Detail Level statement. XRDELETE must begin in column 6. Only one member may be coded per XRDELETE statement but multiple XRDELETE statements are allowed.

#### DSS0825 SEG OVERFLOW. CNT = cnt . ITEM = item

Explanation: Segment table overflow. DOCU/TEXT provides for a segment count of 100 for each item. The count was exceeded for the specified member.

Action: Contact ASG Customer Support.

#### DSS0826 NUMBER OF RECORDS PROCESSED WAS count

Explanation: Lists the number of records processed during an SDF Group Copy. The count is the number of records written to the second "copy to" file.

Action: Contact ASG Customer Support.

#### DSS0827 DB2 PROCESSING SPECIFIED WITHOUT ACC, CAC, OR VTOC

Explanation: DB2 control cards reside in a dataset. ACC, CAC and VTOC must be specified in the PARMLIB in order to open and read these cards.

Action: Processing continues as if PPI=NODB2 was specified.

#### DSS0828 WARNING - ATTEMPTED TO WRITE DUPLICATE KEY

Explanation: A warning message is issued when there's an attempt to write a duplicate ONLINE XREF key.

Action: The record key is given in warning message DSS0829.

#### DSS0829 KEY: xxxxx

Explanation: A warning message is issued showing the record key related to the DSS0828 warning message -- duplicate ONLINE XREF key.

Action: Contact ASG Customer Support.

#### DSS0830 OPEN ERROR-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 99

Explanation: ONLINE XREF encountered an error opening the JLLTM file. The VSAM return code and error text are shown.

Action: Contact ASG Customer Support.

#### DSS0831 INCREASE VSAM FILE PRIMARY SPACE, RESTORE AND RERUN

Explanation: ONLINE XREF encountered an error writing the JLLTM file.

Action: Reallocate the dataset with more primary space, restore the file from backup and re-run the update.

#### DSS0832 LAST VSAM OPERATION WAS: XXXXXXXXXXXXXXX

Explanation: ONLINE XREF identifies the last VSAM operation and its location in the program.

Action: Contact ASG Customer Support.

#### DSS0833 DISCARDED:XXXXXXXXXXXXXXXXXXXXXXXXXXXX

Explanation: ONLINE XREF encountered a record that could not be processed.

Action: The index numbers, job name, procedure name, and program name will indicate which record was discarded.

#### DSS0840 ZARA INITIALIZATION ERROR. RC = xxxx

Explanation: ZARA initialization did not complete correctly. The return code from ZARA initialization attempt is shown.

Action: The ZARA subsystem is probably not up or is on the way down.

#### DSS0841 ZARA API MODULE ZARAAPI1 NOT AVAILABLE.

Explanation: The ZARA application program interface module is not available to JOB/SCAN.

Action: Check that the ZARAAPI1 program is in a library accessible to this run of JOB/SCAN.

#### DSS0842 ZARA API ABEND SYSTEM - sss USER - Unnnn

Explanation: The ZARA application program interface module has abended with the shown abend code and return code. ZARA processing is terminated.

Action: Contact ASG Customer Support.

#### DSS0843 CANNOT CONVERT ZARA DATE FOR dsname

Explanation: D00YZARA is unable to convert the creation date for the shown dataset name to printable format.

Action: Contact ASG Customer Support.

**DSS0844 ZARA API PLIST INVALID. D00YZARA INTERNAL ERROR.**

Explanation: An invalid parameter list has been passed to the ZARA API program ZARAAPI1.

Action: Contact ASG Customer Support.

**DSS0845 CANNOT OPEN VMF FILE.**

Explanation: The TLMS Volume Master File failed to open.

Action: Check the job log for other message relating to the VMF.

**DSS0846 CANNOT IDENTIFY VMF FILE**

Explanation: The dataset that was allocated to the JTLMSVMF DD statement did have the necessary header information to identify the TLMS Volume Master File.

Action: Make sure the correct dataset name is specified.

**DSS0852 VTOC READ FOR xxx FAILED - VOLUME IOERR ON xxx**

Explanation: An I/O error indicator was returned from a VTOC read. Possible hardware problems on the specified volume.

Action: Consult your systems programmer.

**DSS0853 CANNOT OPEN CA1 - CATALOG - OPTION REMOVED**

Explanation: The open request to TMS was rejected.

Action: The TMS option is ignored for the remainder of the analysis.

**DSS0854 CANNOT CONVERT TMS DATE FOR x**

Explanation: A date returned from TMS failed standard date conversion.

Action: Consult your tape librarian.

**DSS0856 TMS MODULE TMSSVC NOT AVAILABLE**

Explanation: The TMS interface module TMSSVC was not found.

Action: Refer to [CA-1 \(UCC1\) Tape Management System](#) for more information.

**DSS0857 TMS Y SVC ABEND xyy-rc**

Explanation: This message is written when an ABEND is detected in the TMS Y SVC. It is equivalent to the TMS IEFTMS70 message issued by CA1. 'xyy' represents the CA1 ABEND code and user Y SVC number. 'rc' is the return code.

Action: Refer to the CA1 manual that describes the IEFTMS70 message for more information.

**DSS0860 CALL ATTACH FACILITY (CAF) MODULE DSNALI NOT FOUND**

Explanation: The CALL ATTACHMENT FACILITY (CAF) interface to DB2 requires a module named "DSNALI". This module could not be located.

Action: Add the appropriate DB2 load library name to the JOBLIB/STEPLIB DD concatenation and run the job again.

**DSS0861 DB2 subsystem CONNECT REQUEST FAILED**

- Explanation: The CALL ATTACHMENT FACILITY (CAF) interface to DB2 attempted to CONNECT to DB2 and the CONNECT was not successful.
- Action: This message is usually followed by a more specific message describing the reason for the CONNECT failure.

**DSS0862 ALREADY CONNECTED TO DB2 subsystem**

- Explanation: The CALL ATTACHMENT FACILITY (CAF) interface to DB2 received a CONNECT request but was already connected to another DB2 subsystem.
- Action: A DISCONNECT from the current DB2 subsystem is then attempted.

**DSS0863 DISCONNECT FROM DB2 subsystem FAILED**

- Explanation: The CALL ATTACHMENT FACILITY (CAF) interface to DB2 attempted to DISCONNECT from the DB2 subsystem and the DISCONNECT was unsuccessful.
- Action: Contact ASG Customer Support.

**DSS0864 DB2 subsystem OPEN REQUEST FAILED FOR PLAN planname**

- Explanation: The CALL ATTACHMENT FACILITY (CAF) interface to DB2 attempted an OPEN (allocate) for the named plan and the OPEN was unsuccessful.
- Action: Contact ASG Customer Support.

**DSS0865 DB2 subsystem ALREADY HAS PLAN planname ALLOCATED**

- Explanation: The CALL ATTACHMENT FACILITY (CAF) interface to DB2 attempted an OPEN (allocate) for a plan name and another plan name is currently allocated. A CLOSE for the current plan name is attempted.
- Action: Contact ASG Customer Support.

**DSS0866 DB2 subsystem CLOSE FAILED FOR PLAN planname**

- Explanation: The CALL ATTACHMENT FACILITY (CAF) interface to DB2 attempted a CLOSE (deallocate) for a plan name and the CLOSE was not successful.
- Action: Contact ASG Customer Support.

**DSS0867 DB2 subsystem DISCONNECT REQUEST FAILED**

- Explanation: The CALL ATTACHMENT FACILITY (CAF) interface to DB2 attempted to DISCONNECT from DB2 and the DISCONNECT was not successful.
- Action: Contact ASG Customer Support.

**DSS0868 CAF RETURN CODE - returncode REASON CODE - reasoncode**

- Explanation: Return code and reason code for the CALL ATTACHMENT FACILITY (CAF) interface to DB2.
- Action: These codes are explained in the DB2 MESSAGES and CODES manual (section 4).



**DSS0869 DB2 SUBSYSTEM subsystem NOT ACTIVE**

Explanation: The CALL ATTACHMENT FACILITY (CAF) interface to DB2 attempted to CONNECT to the DB2 subsystem and the named subsystem is not currently active.

Action: Contact ASG Customer Support.

**DSS0870 DB2 SUBSYSTEM subsystem NOT FOUND**

Explanation: The CALL ATTACHMENT FACILITY (CAF) interface to DB2 attempted to CONNECT to the DB2 subsystem and the named subsystem could not be located by MVS.

Action: Contact ASG Customer Support.

**DSS0871 NOT AUTHORIZED TO CONNECT TO DB2 SUBSYSTEM subsystem**

Explanation: The CALL ATTACHMENT FACILITY (CAF) interface to DB2 attempted to CONNECT to the DB2 subsystem and you are not authorized to connect to the named subsystem.

Action: Contact ASG Customer Support.

**DSS0872 NOT AUTHORIZED TO ALLOCATE PLAN planname**

Explanation: The CALL ATTACHMENT FACILITY (CAF) interface to DB2 attempted to OPEN (allocate) a PLAN, but you are not authorized to allocate this plan.

Action: Contact ASG Customer Support.

**DSS0873 PLAN planname UNAVAILABLE OR IS NOT KNOWN TO DB2**

Explanation: The CALL ATTACHMENT FACILITY (CAF) interface to DB2 attempted to OPEN (allocate) a PLAN and the named PLAN was either not available or is not known to DB2.

Action: Contact ASG Customer Support.

**DSS0874 MAXIMUM CONNECTIONS - TRY LATER**

Explanation: The CALL ATTACHMENT FACILITY (CAF) interface to DB2 attempted to CONNECT to DB2 but the maximum number of concurrent identify level agents has been exceeded. Attempt connection to DB2 at a later time.

Action: Ensure that the system parameters controlling the maximum number of concurrent identifies are correct.

**DSS0875 DB2 IN RESTRICTED ACCESS MODE - TRY LATER**

Explanation: The CALL ATTACHMENT FACILITY (CAF) interface to DB2 attempted to CONNECT to DB2, however, DB2 had been started in restricted access mode.

Action: Only authorization IDs authorized to perform maintenance functions are permitted to access DB2. Retry the job after DB2 has been restarted in full access mode.

**DSS0876 ACCESS TO DB2 DENIED BY SITE AUTHORIZATION EXIT**

Explanation: The CALL ATTACHMENT FACILITY (CAF) interface to DB2 attempted to CONNECT to DB2, however, your site authorization exit has denied you access.

Action: Contact your security administrator and request authority to access this DB2 subsystem.

#### DSS0880 LOAD OF RMM MODULE EDGXAPI FAILED

Explanation: The IBM RMM Interface module failed to load. The Program Product Interface=RMM has been set to NORMM.

Action: Check with the installer of RMM to make sure the correct libraries have been concatenated to allow access to the RMM API.

#### DSS0881 RMM SUBSYSTEM NOT INITIALIZED

Explanation: The RMM subsystem is not initialized. The PROGRAM PRODUCT INTERFACE=RMM Run Option will be set to NORMM.

Action: Contact ASG Customer Support.

#### DSS0900 OPEN FOR PDS WRITE TO xxx FAILED

Explanation: PDSOUT processing was requested to the named ddname but the output file cannot be opened.

Action: Check the FILE ALLOCATIONS statements coded in PARMLIB. Also, refer to [Installation](#) for detailed information on allocating PDSOUT datasets.

#### DSS0901 MBR xxx NOT WRITTEN TO xxx - DIRECTORY FULL

Explanation: The directory of the specified library was full.

Action: Delete some members from the library or recreate the library, increasing the directory space.

#### DSS0902 ERROR DURING STOW TO xxx - RC=yy-zz

Explanation: An error has occurred while trying to store a member in the directory of the dataset whose DDNAME is xxx. The return code and reason code returned by the STOW SVC (SVC21) is yy and zz, respectively.

Action: See your DASD manager or systems programmer for assistance.

#### DSS0903 DDNAME xxxxxxxx COULD NOT BE OPENED FOR INPUT/OUTPUT

Explanation: The DDNAME "xxxxxxx" could not be opened. Check the JESLOG or system messages dataset of the job or your TSO session for the specific reason the dataset could not be opened.

Action: If you are running TSO, you should have "WTPMSG" specified in your TSO profile.

#### DSS0904 INVALID FILE TYPE FOR zzzz IN LMF - DDNAME xxxxxxxx

Explanation: This is an internal error.

Action: Contact ASG Customer Support.

#### DSS0910 NUMBER OF JOB LIBRARIES EXCEEDED. LIMIT 32

Explanation: Only 32 JOB libraries may be specified in an AIU run.

Action: Any remaining libraries may be specified in another AIU run.

**DSS0911 NUMBER OF PROC LIBRARIES EXCEEDED. LIMIT 32**

Explanation: Only 32 PROC libraries may be specified in an AIU run.  
Action: Any remaining libraries may be specified in another AIU run.

**DSS0912 NUMBER OF DELETED JOBS EXCEEDED. LIMIT 7000**

Explanation: The AIU process has a limit of 7000 jobs that can be deleted in one run.  
Action: The number of jobs being deleted can be reduced by specifying fewer JOB libraries and running the AIU process again. The remaining JOB libraries can be processed in a subsequent AIU run.

**DSS0913 ALLOCATION ERROR: dsn**

Explanation: The listed dsn cannot be allocated.  
Action: Check to see if the dataset name is spelled correctly.

**DSS0914 I/O ERROR MESSAGE**

Explanation: The listed error message will explain what kind of error has occurred.  
Action: Contact ASG Customer Support.

**DSS0915 AIUFLUSH NOT ENABLED. YOU MUST SPECIFY AUTOUPDATE=CRC**

Explanation: AIUFLUSH specifies whether or not to process non-JCL members found in the JOB libraries.  
Action: Contact ASG Customer Support.

**DSS0916 AUTOUPDATE PARM "xxx" NOT VALID. SPECIFY CRC OR STATS.**

Explanation: The AUTOUPDATE parm must be CRC or STATS.  
Action: Contact ASG Customer Support.

**DSS0917 AIUFLUSH PARM "xxx" NOT VALID. SPECIFY YES OR NO.**

Explanation: The AIUFLUSH parm must be YES or NO.  
Action: Contact ASG Customer Support.

**DSS0918 LIBRARY TYPE "xxx" NOT VALID. EXPECTING JOB OR PROC.**

Explanation: The parameter "xxx" cannot be identified.  
Action: Contact ASG Customer Support.

**DSS0930 xxxx ISSUED THE FOLLOWING MESSAGES**

Explanation: The scheduler issued the following message(s).  
Action: Contact ASG Customer Support.

**DSS0931 INCORRECT COMMAND STATEMENT IGNORED**

Explanation: The first word in the command statement is not valid.  
Action: Contact ASG Customer Support.

**DSS0932 ABOVE TEXT BYPASSED TILL NEXT COMMAND FOUND**

Explanation: A valid command was not found and caused text to be by-passed until a valid command was found.

Action: Contact ASG Customer Support.

**DSS0933 TEXT/DATA BYPASSED TILL NEXT STATEMENT FOUND**

Explanation: An error was found in a scheduler statement causing the remainder of the text or data of that statement to be by-passed until the next valid statement was found.

Action: Contact ASG Customer Support.

**DSS0934 REQUIRED xxxx PARAMETER NOT FOUND**

Explanation: The specified parameter has been omitted and is required for the operation requested.

Action: Contact ASG Customer Support.

**DSS0935 EXCESSIVE RIGHT PARENTHESES FOUND**

Explanation: Statement contained more right parentheses than left parentheses.

Action: Contact ASG Customer Support.

**DSS0936 NO VERB FOUND - STATEMENT IGNORED**

Explanation: The scan was looking for a verb but instead received a zero length entry or a blank line.

Action: This might occur if the first character encountered was a comma.

**DSS0937 EXPECTED CONTINUATION NOT RECEIVED**

Explanation: The previous line of JCL implied that it should be continued on the next by the presence of a comma as the last character or a non-blank in column 72, but the next line contained the start of a new statement.

Action: Contact ASG Customer Support.

**DSS0938 COMMENT STARTED WITH "/\*" DOES NOT END WITH "\*/"**

Explanation: JOB/SCAN was unable to find the end of a comment statement within a command stream before encountering the start of the next command.

Action: Contact ASG Customer Support.

**DSS0939 xxxx INCORRECT WITH xxxx**

Explanation: The specified keywords are incompatible.

Action: Contact ASG Customer Support.

**DSS0940 VALUE REQUIRED FOR xxxx KEYWORD**

Explanation: The subparameter list or data required for this keyword is missing.

Action: Contact ASG Customer Support.

**DSS0941 NO VALUE GIVEN FOR VARIABLE**

Explanation: Keyword parameter did not have a required sub-parameter.

Action: Contact ASG Customer Support.

**DSS0942 MISSING RIGHT PARENTHESIS AFTER VALUE**

Explanation: Expected right parenthesis not found where required.

Action: Contact ASG Customer Support.

**DSS0943 MISSING LEFT PARENTHESIS BEFORE VALUE**

Explanation: Expected left parenthesis not found where required.

Action: Contact ASG Customer Support.

**DSS0944 LEFT PARENTHESIS MISSING AFTER KEYWORD**

Explanation: An opening parenthesis delimiting the subparameter list or constants associated with the keyword does not precede the keyword.

Action: Contact ASG Customer Support.

**DSS0945 xxxx IS NOT VALID FOR THIS COMMAND**

Explanation: Keyword parameter did not have a required sub-parameter.

Action: Contact ASG Customer Support.

**DSS0946 DUPLICATE KEY : xxxx**

Explanation: The same keyword parameter was entered more than once.

Action: Contact ASG Customer Support.

**DSS0947 REQUIRED POSITIONAL PARAMETER NOT FOUND**

Explanation: The command syntax requires a positional parameter.

Action: Contact ASG Customer Support.

**DSS0948 EXCESSIVE NUMBER OF LEFT PARENTHESSES FOUND**

Explanation: Statement contained more left parentheses than right parentheses.

Action: Contact ASG Customer Support.

**DSS0949 "" IS ADDITIONAL POSITIONAL PARAMETER**

Explanation: An additional, illegal positional parameter was found.

Action: Contact ASG Customer Support.

**DSS0950 SUBSYSTEM xxxx NOT ACTIVE**

Explanation: The scheduler is not currently running in the system.

Action: Contact ASG Customer Support.

**DSS0951 BAD DATA RETURNED FROM JCLPREP**

Explanation: The output from JCLPREP is incorrect or incomplete.

Action: Contact ASG Customer Support.

**DSS0952 INPUT JCL DATA SET OR MEMBER IS EMPTY OR INVALID**

Explanation: The dataset or member specified as input JCL contains no data or contains other than EBCDIC data.

Action: Contact ASG Customer Support.

**DSS0953 INVALID ERROR NUMBER x PASSED FROM S8A**

Explanation: Internal diagnostic error.

Action: Contact ASG Customer Support.

**DSS0954 "ACC=Y" RUN OPTION OVERRIDDEN BY "CAC NOVTOC"**

Explanation: "CATALOG ACCESS NOVTOC" and "ACCESS CONTROL CARDS =YES" were both specified as Run Options in the PARMLIB. "NOVTOC" overrides "ACCESS CARDS =YES".

Action: Therefore, control card datasets will not be listed in the Structured JCL Listing.

**DSS0955 "CAC PDSMEM" RUN OPTION OVERRIDEN BY "CAC NOVTOC"**

Explanation: "NOVTOC" and "PDSMEM" were both specified in the "CATALOG ACCESS" Run Option in the PARMLIB. "NOVTOC" overrides "PDSMEM".

Action: Therefore, PDS members will not be verified.

**DSS0956 RETURN CODE = xxxx IN DELETING JLTAZ**

Explanation: Internal diagnostic error.

Action: Contact ASG Customer Support.

**DSS0957 USING DATA IMPORT OPTION - BYPASS JCL PARSING**

Explanation: Standard JCL processing is ignored when TYPRUN=IMPORT is specified.

Action: "IMPORT" should only be specified when performing the ONLINE XREF Data Import function.

**DSS0958 USING DATA IMPORT OPTION - CREATE PARSE FILE**

Explanation: Stops processing after the JCL has been parsed when the TYPERUN Run Option parameter PARSEONLY=YES is specified.

Action: "PARSEONLY" can be used to increase performance when the same JCL will be parsed for multiple report processing (ie. ONLINE XREF and ONLINE INFO). The File Allocation for the JLAB DD should be changed from the default temporary dataset to a permanent dataset.

**DSS0959 OLX FILE IS CORRUPT - UPDATE NOT PERFORMED**

Explanation: Stops processing before JCL parsing. An error flag was set during a previous OLX update run indicating that the file has been corrupted.

Action: Restore file from a backup copy and re-run the update.

#### DSS0960 BYPASSING OLX FILE CHECK

Explanation: Internal diagnostic message.

Action: Contact ASG Customer Support.

#### DSS0961 XXXXXX - XXXXXXXXXXXXX FORMAT SHOULD BE XXXXXX

Explanation: DSS0961 reports syntax errors on JLSCH input for TWS, CA7, and ESP statements. The first variable field reports which keyword has the syntax error. The second field describes the problem. The third variable field contains the correct format. Example: CA7 SCHID(001) SDATE(100225) STIME(23:59) DSS0961E - STIME - TOO LONG FORMAT SHOULD BE HHMM

Action: Correct the statement in JLSCH input stream and re-submit the job.

#### DSS0962 XXXXXX IS XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Explanation: DSS0962 reports syntax errors on JLSCH input for TWS, CA7, and ESP statements. The first variable field reports which keyword has the syntax error. The second field describes the problem. Example: TWS ADID(APPL01) OPRNUM(256) - DSS0962E - OPRNUM IS GREATER THAN 255

Action: Correct the statement in JLSCH input stream and re-submit the job.

#### DSS0980 JOBTRAC INTERFACE PROCESS

Explanation: The JOBTRAC interface process has started.

Action: This is an informational message from the JOBTRAC interface processor.

#### DSS0981 nnnn RECORDS READ FROM FORECAST FILE

Explanation: The JOBTRAC interface processor has read 'nnnn' records from the JOBTRAC REPORTER "DISPLAY SCHEDULE LISTING".

Action: This is an informational message from the JOBTRAC interface processor.

#### DSS0982 nnnn SELECT RECORDS WRITTEN

Explanation: The JOBTRAC interface processor has written 'nnnn' job names to be scanned or documented.

Action: This is an informational message from the JOBTRAC interface processor.

#### DSS0983 DDNAME xxxxxxxx NOT FOUND

Explanation: The JOBTRAC interface processor was unable to locate the DD named 'xxxxxxx'.

Action: The DD name must be specified in a previously allocated JCL statement or specified in a 'FILE ALLOCATIONS' statement in a PARMLIB library member. The two DD names that are required by the JOBTRAC interface processor are: FORECAST - The JOBTRAC REPORTER "DISPLAY SCHEDULE" LISTING that is the output of the CA-JOBTRAC program "JREPORT3".  
 JIJOB - The dataset or dataset concatenation that specifies the library or libraries that contain the jobs to be scanned or documented.  
 JCL MESSAGES =====

**DSS1010 INPUT STREAM CONTAINS NO JOBS**

**Explanation:** The JCL stream to be scanned contains no input. This may be caused by misspelling the member name, by accessing the wrong input library, or by selecting from a library that contains PROCs whose library type was not specified as PROCLIB.

**Action:** If you are using ISPF JOB/SCAN or DOCU/TEXT you must specify one or more JCL members for processing in the Select Members field or Select List field of the Execution or Create panels or Directory sub-panels. Verify that the library is the correct one by looking at the entry in the Input Lib or List Lib field. To change the Library, overwrite the field or access the Library Specification panel. If you are using batch submission, check that INPUT SELECTION statements are entered and spelled correctly. These statement types are:  
 SELECT Specifies JCL member names (ALL selects all members)  
 EXCLUDE Excludes specific members from among those selected  
 DSN Identifies library to be accessed  
 DDNAME Identifies ddname of library to be accessed SOURCE TYPE Identifies type of library (PAN, LIB, LIB-3, PROC or sequential - Default is PDS) Example: INPUT SELECTION DSN=N=SYS2.JOBS SELECT=JOB1-JOB99 Select a range of jobs range1-range2 SELECT=PREFIX=PAY Select all jobs beginning with certain chars  
 EXCLUDE=JOB50 Select jobs fitting a mask (ex. JOB\*\*UPD) SELECT=ALL Selects all jobs in a system

Repeat INPUT SELECTION Group Level statement and DSN and SELECTs to specify multiple libraries.

**DSS1011 MEMBER SKIPPED - VALID JCL NOT FOUND**

**Explanation:** The member being scanned did not start with a JCL statement

**Action:** Contact ASG Customer Support.

**DSS1014 COMMENT STATEMENT FOUND BEFORE JOB OR PROC STATEMENT**

**Explanation:** JOB/SCAN found one or more comment statements in the ISPF EDIT work area before a JOB statement or a PROC statement was found and will be ignored (not processed) by the JEM ISPF EDIT macro.

**Action:** Contact ASG Customer Support.

**DSS1015 IMBEDDED NULL (//) FOUND WITHIN JCL STREAM**

**Explanation:** The JCL stream to be scanned contains a null // statement not followed by a legal JOB card. MVS will flush all JCL until the next legal JOB card is found.

**Action:** All JCL flushed is noted in cc73-80 with the text 'FLUSHED' and not processed by DOCU/TEXT or JOB/SCAN.

**DSS1016 JCL FOUND BEFORE FIRST VALID JOB CARD IN MEMBER ( )**



Explanation: The JCL stream to be scanned contains JCL before the first legal JOB statement. MVS will flush all JCL until the next legal JOB statement is found.

Action: All JCL flushed is noted in cc73-80 with the text 'FLUSHED' and not validated by JOB/SCAN or documented by DOCU/TEXT.

Note: If multiple PDS members are scanned, and RUN OPTION SPANNED MEMBERS = NO is coded (or defaulted), each individual members' JCL must also contain a valid JOB card prior to the JCL (excepting certain JES2 and JES3 statements).

#### DSS1017 MEMBER SKIPPED - "/" NOT FOUND IN COL 1.

Explanation: The member being scanned did not have a / in column 1 of the first line processed.

Action: Contact ASG Customer Support.

#### DSS1018 EMBEDDED NULL (//) FOUND FLUSHING TO END OF PROC

Explanation: The procedure being validated contained a null card. The rest of the procedure is flushed.

Action: Validation resumes after the end of the procedure.

#### DSS1019 EMBEDDED NULL (//) FOUND FLUSHING TO END OF INCLUDE

Explanation: The include being validated contained a null card. The rest of the include is flushed.

Action: Validation resumes after the end of the include.

#### DSS1020 VALUE OF OUTLIM PARAMETER NOT NUMERIC

Explanation: Related IBM message - IEF644I

Action: The OUTLIM parameter, if coded, must have a numeric value between 1 and 16777215. For JES2, if OUTLIM is not coded, there is no limit. JES3 uses site-defined limit.

#### DSS1025 Value of OUTLIM parameter exceeds 16777215

Explanation: Related IBM message - IEF638I

Action: The OUTLIM parameter, if coded, must have a numeric value between 1 and 6777215.

#### DSS1030 JOB ADDRSPC IS NOT "VIRT" OR "REAL"

Explanation: The ADDRSPC parameter of the JOB statement, if coded, must be either "VIRT" or "REAL".

Action: VIRT (usually the default) requests virtual storage and allows job or step to be paged. REAL requests central or real storage and the system cannot page.

#### DSS1035 ACCTNG FIELD CONTAINS SPECIAL CHAR - QUOTES NEEDED

Explanation: The ACCOUNTING field of the JOB statement contains a special character which must be enclosed in single quotes.

Action: Blank,./)(\*&=+ must be enclosed in single quotes.

#### DSS1037 INVALID ACCOUNTING FIELD IN THE JOB STATEMENT

Explanation: The ACCOUNTING field of the JOB statement contains a syntax error.

Action: Check for missing apostrophes, missing commas, etc. Consult the IBM JCL manual for the correct syntax of this parameter. (acct-#)(pa#,room,time,lines,cards,forms,copies,log,linect) Code a comma in place of each omitted parameter when other parameters follow. Max length is 142 characters. Subparameters containing special characters (blank,./')(\*&=+) must be enclosed in single quotes. Apostrophes and ampersands must appear as double characters.

#### DSS1040 JOB CLASS IS LONGER THAN ONE CHARACTER OR MISSING

Explanation: The CLASS parameter of the JOB statement must have a ONE-character value of A-Z or 0-9.

Action: Contact ASG Customer Support.

#### DSS1041 JOB CLASS VALUE IS NOT A THRU Z OR 0 THRU 9

Explanation: The CLASS parameter of the JOB statement must have a ONE-character value of A - Z or 0 - 9.

Action: Contact ASG Customer Support.

#### DSS1045 MSGCLASS IS LONGER THAN ONE CHARACTER OR IS MISSING

Explanation: The MSGCLASS parameter of the JOB statement must have a ONE-character value of A-Z or 0-9.

Action: Contact ASG Customer Support.

#### DSS1046 MSGCLASS VALUE IS NOT A THRU Z OR 0 THRU 9

Explanation: The MSGCLASS parameter of the JOB statement must have a ONE-character value of A-Z or 0-9.

Action: Contact ASG Customer Support.

#### DSS1050 NON-NUMERIC OR INVALID DATA IN JOB PRTY PARAMETER

Explanation: The PRTY parameter of the JOB statement must contain a numeric value of 0-15 for JES2 or 0-14 for JES3.

Action: Contact ASG Customer Support.

#### DSS1051 JOBNAME MISSING ON THE JOB STATEMENT

Explanation: A JOB name must be coded on the JOB JCL statement. MVS will flush a job with no name on the JOB statement.

Action: Contact ASG Customer Support.

#### DSS1052 VALUE OF IS INCORRECT IN JOB STATEMENT

Explanation: &SYSUID coded in the specified JOB statement parameter is incorrect.

Action: If JCLERR=YES is coded in the JES2 initialization parameter JOBDEF, then the job will be terminated by JES2. If JCLERR=NO is coded, then this message can be ignored or suppressed.

#### DSS1054 CCSID MUST BE A DECIMAL NUMBER BETWEEN 1 AND 65535

Explanation: The coded character set identifier (CCSID) must be specified as a decimal number between 1 and 65535.

Action: Contact ASG Customer Support.

**DSS1060 RD VALUE IS NOT "R", "RNC", "NC", OR "NR".**

Explanation: Related IBM message - IEF643I

Action: The RD parameter of the JOB statement must contain either "R", "RNC", "NC", or "NR".

**DSS1065 COLUMNS 68-71 OF JOB STATEMENT ARE NOT BLANK (CA-7)**

Explanation: In your JCL, columns 70 and 71 of the JOB statement must be blank for any job to be submitted by CA-7.

Action: If external security with USERID insertion is used, column 69 must be a blank or a comma. If CA-7 NCF is used, columns 69, 70, and 71 of the JOB statement must be blank for any job to be submitted by CA-7. If external security with USERID insertion is used, column 68 must be a blank or a comma.

**DSS1070 NON-NUMERIC OR INVALID VALUE IN TIME PARAMETER**

Explanation: The TIME parameter of the JOB statement must be a numeric value not greater than 1440.

Action: Format is TIME=(min,sec) or TIME=min or TIME=(,sec). example, TIME=(1,30); TIME=5; TIME=(,30); TIME=1440

**DSS1078 INCORRECT JESLOG PARAMETER VALUE**

Explanation: An incorrect value was coded for the JESLOG parameter in the JOB statement.

Action: Format is JESLOG=SPIN or JESLOG=NOSPIN or JESLOG=SUPPRESS or JESLOG=(SPIN,nnn) or JESLOG=(SPIN,nnnK) or JESLOG=(SPIN,nnnM) or JESLOG=(SPIN,'hh:mm') or JESLOG=(SPIN,'+hh:mm'). Check for missing parentheses, missing apostrophes, less than 500 lines, hours greater than 23, minutes greater than 59, time interval less than 10 minutes or misspelled keyword.

**DSS1079 INCORRECT JOBRC PARAMETER VALUE**

Explanation: An incorrect value was coded for the JOBRC parameter in the JOB statement.

Action: Format is JOBRC=MAXCC or JOBRC=LASTCC or JOBRC=(STEP,stepname) or JOBRC=(STEP,stepname.procstepname).

**DSS1080 INVALID JOB PARAMETER - " "**

Explanation: The JOB statement contains an invalid or incorrectly spelled keyword.

Action: Valid keywords, which are not standard IBM keywords, may be listed in Control Table AZ to avoid DSS1080 errors.

**DSS1085 INVALID RESTART PARAMETER**

Explanation: The parameter(s) coded on the RESTART statement are not valid.

Action: Verify their contents and/or see the IBM manual for the correct format for coding this parameter. RESTART=(\*,checkid) no parentheses if no 2nd subparameter stepname stepname,procstepname

**DSS1090 INVALID JOB PARAMETER FORMAT - NO EQUAL SIGN**

Explanation: The JOB statement has an invalid format, sequence of parameters, or missing equal sign (=).  
Action: Verify the accounting fields for proper ordering and make sure other JOB statement parameters are followed by an equal sign (=).

#### DSS1095 PASSWORD VALUE GREATER THAN 8 CHARACTERS

Explanation: The value for the PASSWORD parameter on the JOB statement cannot exceed eight characters.  
Action: Contact ASG Customer Support.

#### DSS1100 JOB HAS NO STEPS

Explanation: Related IBM message - IEF607I or IEF607I  
Action: The JCL contains no valid EXECUTE statement. It must contain one of the following: // EXEC PGM=abc // EXEC PROC=abc // EXEC procname

#### DSS1110 PROC procname HAS NO STEP

Explanation: Related IBM message - IEF610I or IEF610I  
Action: The scan did not find an EXEC statement between the PROC statement and the end of the procedure. A misplaced PROC or PEND statement or a missing EXEC statement is usually the cause.

#### DSS1112 ISPF PACKED DATA IS INVALID FOR PROCLIB

Explanation: The member named in the EXEC PROC= statement or MEMBER= parm of the INCLUDE statement contains data that is possibly in the ISPF EDIT 'PACK ON' format.  
Action: Edit the member and enter the 'PACK OFF' ISPF EDIT profile command. Save the member and resubmit the JOB.

#### DSS1115 NO PEND CARD FOUND FOR INLINE PROC

Explanation: An instream PROC must begin with a PROC statement and must end with a PEND statement.  
Action: The format is: //label PEND comments

#### DSS1120 PROGRAMMER NAME IS LONGER THAN ALLOWED

Explanation: The Programmer Name field in the JOB statement contains more than 20 characters.  
Action: When the Programmer Name field is enclosed in apostrophes, the 20 characters do not include the enclosing apostrophes.

#### DSS1122 INVALID CHARACTER IN PROGRAMMER NAME

Explanation: The Programmer Name field must be enclosed in apostrophes when it contains special characters other than hyphens, leading periods, or embedded periods.  
Action: Contact ASG Customer Support.

#### DSS1125 JOB STATEMENT INVALID IN PROCEDURE

Explanation: JOB statements are not allowed in cataloged procedures.  
Action: Contact ASG Customer Support.

**DSS1126 JES2/JES3 STATEMENT INVALID IN PROCEDURE**

Explanation: JES2/JES3 statements are not allowed in cataloged procedures.

Action: Contact ASG Customer Support.

**DSS1127 INCORRECT JCL STATEMENT IN PROCEDURE**

Explanation: JOB/SCAN detected an incorrect JCL statement in a cataloged or an instream procedure.

Action: Examples of an incorrect JCL statement are a '/'\* ' delimiter statement, a blank statement, a control statement, etc.

**DSS1129 INVALID VALUE FOR SECLABEL**

Explanation: The value of the SECLABEL parameter in the JOB statement is invalid.

Action: The SECLABEL name must be 1-8 alphanumeric or national (\$,#,@) characters. The first character must be alphabetic or \$, #, @.

**DSS1130 INVALID VALUE FOR TYPRUN**

Explanation: The value of the TYPRUN (JOB level) keyword was not SCAN, HOLD, COPY, or JCLHOLD. Verify the spelling of the value.

Action: Contact ASG Customer Support.

**DSS1135 INVALID VALUE FOR USER**

Explanation: Related IBM message - IEF647I

Action: The value of the USER (JOB level) keyword was not alphanumeric or did not start with alphabetic or national (\$,#, @) characters. Length is 1 - 7 characters.

**DSS1136 "USER" VALUE LONGER THAN 8 CHARACTERS OR MISSING**

Explanation: The value of the USER (JOB level) keyword was more than 8 characters long or not found.

Action: Contact ASG Customer Support.

**DSS1137 DLM= REQUIRES TWO CHARACTERS. FOUND " "**

Explanation: Related IBM message - IEF632 Format Error in the DLM field

Action: The value of the DD card DLM= subparameter has an incorrect length. Both JES2 and JES3 require two (2) DLM characters. Any other value will cause JES2 or JES3 to fail the job.

**DSS1138 Delimiter not found**

Explanation: A delimiter defined in a DLM=XX DD Statement was not found prior to the end of the member.

Action: Contact ASG Customer Support.

**DSS1140 NON-NUMERIC DATA IN PERFORM**

Explanation: The value of PERFORM in the JOB statement contains non-numeric data.

Action: Contact ASG Customer Support.

**DSS1145 "PERFORM" VALUE LONGER THAN 3 DIGITS OR MISSING**

**Explanation:** The value of PERFORM in the job statement contains too many digits or is not present. Valid values are 1 thru 999.

**Action:** Contact ASG Customer Support.

**DSS1150 NOTIFY PARAMETER CONTAINS TOO MANY CHARACTERS**

**Explanation:** Related IBM message - IEF642I The value of the 'NOTIFY=' JOB parameter is too long.

**Action:** The value of the parameter for JES2 systems must be in one of the following two formats: node-name.userid Nodename is 1 through 8 alphanumeric or national (\$,#,@) characters. Userid is 1 through 8 alphanumeric or national (\$,#,@) characters. userid Userid is 1 through 7 alphanumeric or national (\$,#,@) characters. The value of the parameter for JES3 systems must be: userid Userid is 1 through 7 alphanumeric or national (\$,#,@) characters.

**DSS1151 SCHENV PARAMETER GREATER THAN 16 CHARACTERS LONG**

**Explanation:** The value of the 'SCHENV=' JOB parameter cannot be greater than 16 characters long.

**Action:** This value should be the name of a valid WLM scheduling environment.

**DSS1152 SCHENV PARAMETER CANNOT END WITH AN '\_'**

**Explanation:** The value of the 'SCHENV=' JOB parameter cannot be end with an underscore.

**Action:** Underscores must be imbedded (for example, PLEX\_D01 is valid bit PLEX\_ is not).

**DSS1153 SCHEDULING ENVIRONMENT IS NOT AVAILABLE**

**Explanation:** The scheduling environment named in the SCHENV parameter is not available on the current system.

**Action:** Contact ASG Customer Support.

**DSS1154 SCHEDULING ENVIRONMENT WAS NOT FOUND**

**Explanation:** The scheduling environment named in the SCHENV parameter is not defined to the current system.

**Action:** Contact ASG Customer Support.

**DSS1160 "name" LIMIT EXCEEDED - PROCESSING ENDED**

**Explanation:** If "name" = JOBLIM, the number of jobs read has reached the limit specified in the 'JOB LIMIT-T=nn' Run Option.

**Action:** Set JOB LIMIT=0 for no limit. If "name" = DEMO, the number of jobs read has reached the limit for the DEMO version of this product.

**DSS1165 "" COUNTER MAX EXCEEDED - PROCESSING CONTINUED**

**Explanation:** After 99998 jobs or 998 jobsteps or 99998 datasets have been processed, normal processing continues.

**Action:** Report index numbers will not be incremented.

**DSS1167E "" IS UNRESOLVED FOR PARMDD**

- Explanation: A PARMDD= parameter was specified on an EXEC statement but the DD name referenced by the PARMDD keyword was not defined in the JOB step.
- Action: The DDNAME specified either is misspelled or the DD statement has not been coded for this STEP.

**DSS1170 "" PARAMETER CODED TWICE IN ""**

- Explanation: The keyword shown has been defined twice. This could occur at either the JOB or EXEC level.
- Action: A value has been defined more than once. Check for duplication.

**DSS1180 RESTART STEP NOT FOUND IN JOB**

- Explanation: The 'RESTART=' JOB parameter specified a step that could not be found in the job.
- Action: Ensure that a correct value of '\*', 'stepname', or 'stepname.procstepname' is specified and the step exists in the job.

**DSS1181 INVALID FORMAT FOR "MEMLIMIT" PARAMETER**

- Explanation: The syntax for the MEMLIMIT parameter is: MEMLIMIT={nnnnnM} | {nnnnnG} | {nnnnnT} | {nnnnnP} | {NOLIMIT} nnnnn may be a value from 0 to 99999. The value may be expressed in megabytes (M), gigabytes (G), terabytes (T), or petabytes (P). NOLIMIT specifies that there is no limit on the virtual pages to be used above the bar.
- Action: Contact ASG Customer Support.

**DSS1185 "MSGLEVEL" VALUE MISSING**

- Explanation: The 'MSGLEVEL' JOB parameter was coded and no value was found. The valid format is '(x,y)'; where x is 0, 1, or 2 and y is 0 or 1.
- Action: Omit parentheses if coding only the first value; 'MSGLEVEL=0'. Code a comma if omitting the first value; 'MSGLEVEL=(,1)'

**DSS1186 INVALID FORMAT FOR "MSGLEVEL" PARAMETER**

- Explanation: The 'MSGLEVEL' job parameter was coded and is not valid.
- Action: The valid format is 'MSGLEVEL=(x,y)'; where x is 0, 1, or 2 and y is 0 or 1. Omit parentheses if coding only the first value; 'MSGLEVEL=0'. Code a comma if omitting the first value; 'MSGLEVEL=(,1)'

**DSS1190 NO PROCESSING DONE STARTING WITH JOB XXXXXXXXXX**

- Explanation: There were more than 99998 jobs coded in the members selected for INFO/X and DOCU/TEXT.
- Action: The job name specified and the subsequent jobs will not be processed.

**DSS1470 ADD\_ENTRY FAILED FOR "exit-name". RC=nn**

- Explanation: An attempt was made to introduce "exit-name" to the LE environment. Frequently RC=12 indicating that "exit-name" was not an LE conforming module. Other return codes (RC=) can be

found in “Language Environment Programming Guide” under “Preinitialization services” for the add\_entry function.

Action: If RC=12, recompile the exit using the exec CCEXIT in JOB/SCAN's SJSSAMP data set.

#### DSS2000 PROC procname EXPANDED FROM text

Explanation: JOB/SCAN found an EXEC statement for a procedure.

Action: Message DSS2002 immediately follows this message if the procedure was found in a library. In the message text: procname - The name of the expanded procedure. text - text is one of the following: PRIVATE - The procedure was first found in a library that was specified in the JCLLIB statement. FIRST PROCLIB - The procedure was first found in a library that was specified in a JLJPC DD statement. JLJPA - The procedure was first found in a library that was specified in a JLJPA DD statement. INSTREAM PROC DEFINITION - The procedure was first found in the JCL stream.

#### DSS2001 INCLUDE GROUP group-name EXPANDED FROM text

Explanation: JOB/SCAN found an INCLUDE statement to include a group of JCL statements.

Action: Message DSS2002 immediately follows this message. In the message text: group-name - The name of the included group. text - text is one of the following: PRIVATE - The include group was first found in a library that was specified in the JCLLIB statement. FIRST PROCLIB - The include group was first found in a library that was specified in a JLJPC DD statement. JLJPA - The include group was first found in a library that was specified in a JLJPA DD statement.

#### DSS2002 LIBRARY dsname

Explanation: This message is a continuation of messages DSS2000 and DSS2001.

Action: In the message text 'dsname' is the name of the library where the procedure (DSS2000) or the group-name (DSS2001) was first found.

#### DSS2008 I/O ERROR READING PROCLIB DEVICE

Explanation: Related IBM message - IEF603I An I/O error occurred while reading the JCLLIB concatenation.

Action: A fixed length dataset (RECFM=F) was specified first in the ORDER= parameter followed by a blocked dataset (RECFM=FB) with a BLKSIZE larger than the first dataset.

#### DSS2010 OVERRIDDEN STEP " " NOT FOUND IN PROCEDURE

Explanation: Related IBM message - IEF611I or IEF611I

Action: The step name specified in the override was not found in the PROC. Either an override statement was placed out of order or the step name was misspelled.

#### DSS2020 PROCEDURE " " NOT FOUND

Explanation: Related IBM message - IEF612I or IEF612I IEF663I or IEF663I

Action: The procedure specified in the EXEC PROC name statement could not be found in the job stream or any of the PROCLIBs made available to JOB/SCAN or DOCU/TEXT. Check the name on



the EXEC statement for spelling errors and make sure the proper PROCLIBs are included.  
 JOB/SCAN searches for PROCs in the following order:  
 1) instream PROCs  
 2) the PROCLIB specified in JOB/SCAN Library Specification panel  
 3) //PROCLIB DD in the job (when PPI=EZPROCLIB Run Option is used)  
 4) JIIPA DDname specified as in the File Allocation section in the PARMLIB.

#### DSS2022 RECURSIVE EXECUTION OF PROCEDURE procname

Explanation: Processing of this EXEC statement stops.  
 Action: This EXEC statement executes the named procedure indefinitely.

#### DSS2030 OVERRIDE STATEMENT DOES NOT FOLLOW EXEC PROC

Explanation: The override statement should follow an EXEC PROC name statement.  
 Action: Either the EXEC PROC name is missing or misplaced or the override statement is misplaced.

#### DSS2040 SYMBOLIC NAME " " LONGER THAN 7 CHARACTERS

Explanation: Related IBM message - IEF642I  
 Action: Symbolic names used in PROCs cannot be longer than 7 characters following the "&". e.g. &SYMBNAME.

#### DSS2045 SYMBOLIC NAME " " LONGER THAN 8 CHARACTERS

Explanation: Symbolic names used in the job stream and on cataloged or instream procedures cannot be longer than 8 characters following the "&". e.g. &SYMBNAME.  
 Action: Contact ASG Customer Support.

#### DSS2048 Missing Program Name

Explanation: The EXEC statement contains a null or blank value for the PGM= parameter.  
 Action: Contact ASG Customer Support.

#### DSS2050 PROGRAM " " NOT FOUND

Explanation: Related IBM message - ABEND S806-04  
 Action: The program specified in the EXEC PGM= statement could not be found in the JOBLIB/STEPLIBs, the system LPALIB, or the system link list. Check the spelling of the program name and make sure the appropriate library actually contains the program.

#### DSS2051 LIBRARY NOT SEARCHED - ALLOCATION OR OPEN FAILURE

Explanation: This message follows the DSS2050 warning message.  
 Action: A STEPLIB or JOBLIB dataset in the JCL has failed to allocate or if allocated, has failed to open. The program name specified may be in this dataset. This message could be caused by an exclusive ENQ on the dataset by another job or user at the time of the JOB/SCAN run.

#### DSS2052 LIBRARY NOT SEARCHED FOR PROGRAM NAME

- Explanation: The STEPLIB or JOBLIB dataset in the preceding statement may have been a temporary dataset or an error was detected while parsing the statement and the library could not be allocated by JOB/SCAN.
- Action: The search for the program will continue with the next dataset in the concatenation list or will proceed with the system link pack area (LPALIB) and the system link library (LNKLST).

**DSS2053 LIBRARY NOT SEARCHED FOR MEMBER NAME**

- Explanation: The named member was not validated. Check validity of library name.
- Action: Contact ASG Customer Support.

**DSS2054 Library not valid for ""**

- Explanation: The library specified is invalid for a PROCLIB, JOBPROC, DYNAPROC DD or a JCLLIB statement.
- Action: The library should not be a temporary dataset, a partitioned dataset with a member specified or a generation dataset.

**DSS2056 Library not searched for PROCEDURES or INCLUDE Mbrs**

- Explanation: The PROCLIB, JOBPROC, DYNAPROC or JCLLIB dataset in the preceding statement was either a temporary dataset, a partitioned dataset with a member specified, a GDG, or an error was detected while parsing the library name.
- Action: The library could not be allocated by JOB/SCAN. The search for procedures or INCLUDE members will continue with the next dataset in the concatenation list.

**DSS2060 STEP "" IN "" NOT IN PROC**

- Explanation: The STEP name specified in a COND parameter was not found in the PROC.
- Action: Check for misspelled step names.

**DSS2070 ERROR IN STATEMENT NAME**

- Explanation: This format error is usually the result of a missing space after the statement name.
- Action: Make sure that the statement name is no longer than 8 characters and that there is a space following it.

**DSS2080 PEND CARD FOUND IN CATALOGED PROCEDURE**

- Explanation: The PEND card is not allowed in a cataloged procedure.
- Action: A possible cause may be that an instream PROC was copied into the PROCLIB without removing the PEND card. Remove the PEND card from the cataloged procedure.

**DSS2085 "PEND" INVALID OUTSIDE INSTREAM PROCEDURE**

- Explanation: The PEND card is not allowed within a JCL stream.
- Action: A possible cause may be that an instream PROC was moved into a PROCLIB and the PEND card was not removed from the JCL. Remove the PEND card from the JCL. Also, two PEND cards in an instream PROC will cause this error.

**DSS2087 STATEMENT FOLLOWS "PEND" IN A CATALOGED PROCEDURE**

Explanation: A misplaced PEND statement causes this error.  
Action: This error is caused when a JCL statement follows the PEND card.

**DSS2089 EXTRANEIOUS "PEND" STATEMENT FOUND IN CATALOGED PROCEDURE**

Explanation: A misplaced PEND statement causes this error.  
Action: Contact ASG Customer Support.

**DSS2090 ERROR IN SYMBOLIC NAME**

Explanation: A symbolic must be followed by an equal sign (=) and then the value.  
Action: If the value is to be null, the equal sign (=) still must be coded followed by a comma unless it is the last parameter on that statement.

**DSS2093 "" NOT PROPERLY DELIMITED - MISSING EQUAL**

Explanation: A symbolic must be followed by an equal sign (=) and then the value.  
Action: If the value is to be null, the equal sign (=) still must be coded followed by a comma. If this is the last symbolic to be coded, the comma is not needed but the equal (=) sign is. This error can also be caused by comments on a PROC statement that do not have any symbolics coded. Comments are only allowed if there is at least one symbolic coded first.

**DSS2095 ERROR IN SYMBOLIC NAME FORMAT ""**

Explanation: A symbolic name must be 1 to 8 characters in length and begin with an alphabetic or national ('\$','#','@') character.  
Action: The remaining seven characters can be alphanumeric or national.

**DSS2100 INVALID SYMBOLIC FORMAT**

Explanation: A symbolic value must immediately follow the equal sign (=) without any spaces between.  
Action: Contact ASG Customer Support.

**DSS2110 VALUE OF SYMBOLIC NOT PROPERLY DELIMITED**

Explanation: The value being assigned to the symbolic name extended to (or past) column 72.  
Action: Contact ASG Customer Support.

**DSS2115 VALUE "" MUST BE IN QUOTES**

Explanation: If the value to be assigned to a symbolic contains special characters, they must be enclosed in apostrophes.  
Action: The enclosing apostrophes are not considered part of the value.

**DSS2116 QUOTES NOT ALLOWED FOR "" VALUE**

Explanation: The value coded for the noted parameter must not be enclosed in quotes.

Action: Quotes must be removed before running the job. See the IBM JCL Reference Manual for further information.

#### DSS2117 LOCATION OF QUOTE NOT CORRECT FOR " "

Explanation: A quote was found within a symbolic value.

Action: If quotes are to be part of the value, the entire value must be enclosed in quotes and the internal quote must be coded twice. This message may also be caused by quotes that are not outside the entire symbolic value.

#### DSS2120 VALUE OF SYMBOLIC EXCEEDS INTERNAL LIMIT

Explanation: The maximum length that can be assigned to any symbolic is 63 characters.

Action: Contact ASG Customer Support.

#### DSS2130 PROC CALL WITHIN A PROC NOT ALLOWED

Explanation: Related IBM message - IEF613I

Action: JCL syntax rules do not allow an EXEC PROC statement within another PROC.

#### DSS2140 INSTREAM DATA NOT ALLOWED IN JES3 PROC

Explanation: JCL syntax rules do not allow the use of DD \* or DD DATA within a JES3 PROC.

Action: You can get around this by using DD DUMMY in the PROC and then override that DD by a DD \* or DD DATA followed by the data cards at execution time.

#### DSS2150 WARNING, " " DOUBLY DEFINED IN EXEC PROC

Explanation: The symbolic has been assigned a value more than once in the EXEC statement invoking a PROC.

Action: While this is not a true error, it may lead to unpredictable results at run time. Remove the one that is not needed.

#### DSS2160 WARNING, " " DOUBLE DEFINED IN PROC

Explanation: The symbolic has been assigned a value more than once in the PROC statement.

Action: This will lead to unpredictable results. Remove one of the values.

#### DSS2170 SYMBOLIC " " NOT RESOLVED

Explanation: The symbolic that occurs in a JCL statement within a PROC has not been assigned a value either from the EXEC statement or from the PROC statement as a default.

Action: Contact ASG Customer Support.

#### DSS2175 " " NOT ALLOWED AS SYMBOLIC NAME

Explanation: EXEC statement parameter and subparameter keywords cannot be used as a symbolic parameter name.

Action: Contact ASG Customer Support.

**DSS2180 EXEC SYMBOLIC VARIABLE "" NOT USED IN PROC**

- Explanation: The EXEC PROC statement specified a symbolic variable that did not exist in the PROC.
- Action: This could be the result of a misspelled symbolic name, statements deleted from a PROC after it was created, etc.

**DSS2190 PROC SYMBOLIC VARIABLE "" NOT USED IN PROC**

- Explanation: Related IBM message - IEF657I or IEF657I
- Action: The PROC statement specified a symbolic variable that does not exist in the PROC. This could be the result of a misspelled symbolic name, statements deleted from a PROC after it was created, etc.

**DSS2200 MISSING PEND CARD**

- Explanation: All instream PROCs must end with a PEND card in order to separate the PROC from the rest of the JCL in the job.
- Action: Note that the PEND card CANNOT be used in a cataloged (in a PROCLIB) procedure.

**DSS2201 ONLY ONE SCHEDULE STMT ALLOWED**

- Explanation: More than one SCHEDULE statement was found in a JOB. Only a single SCHEDULE statement is allowed per JOB.
- Action: Remove the extra SCHEDULE statement.

**DSS2202 "WITH" NOT ALLOWED FOR JOBGROUP**

- Explanation: A WITH parameter was found on a JOBGROUP statement. WITH is not valid for JOBGROUP.
- Action: Remove the WITH parameter from the JOBGROUP statement.

**DSS2203 INVALID VALUE FOR JOBGROUP**

- Explanation: An invalid parameter was found on a JOBGROUP statement.
- Action: Consult the "MVS JCL Reference" for valid parameters for the JOBGROUP statement.

**DSS2204 XXXXX IS NOT VALID FOR JOBGROUP**

- Explanation: An incorrect parameter was found on a JOBGROUP statement. Above, XXXXX is replaced by the incorrect parameter.
- Action: Correct the JOBGROUP parameter that indicated in the DSS2204 message. Consult the "MVS JCL Reference" for valid parameters for the JOBGROUP statement.

**DSS2206 XXXXX IS UNDEFINED KEYWORD FOR SCHEDULE**

- Explanation: An incorrect parameter was found on a SCHEDULE statement.
- Action: Consult the "MVS JCL Reference" for valid parameters for the JOBGROUP statement.

**DSS2207 MISSING GROUP NAME FOR SCHEDULE JOBGROUP**

Explanation: A Jobgroup name was found missing from a SCHEDULE JOBGROUP= Statement. A Jobgroup name is required for SCHEDULE JOBGROUP.

Action: Provide a valid Jobgroup name on the SCHEDULE JOBGROUP= Statement.

#### DSS2208 WITH, BEFORE, AFTER NOT OK FOR JOBGROUP

Explanation: One of the above parameters was found on a JOBGROUP statement. Consult the "MVS JCL Reference" for valid parameters for the JOBGROUP statement.

Action: Remove the invalid parameter from the JOBGROUP statement.

#### DSS2210 "" IS OUT OF ORDER

Explanation: Related IBM message - IEF658I or IEF658I

Action: An incorrectly placed statement is embedded in the job stream. This could be caused by not coding a PROC card in a cataloged procedure as the first statement.

#### DSS2250 Subsystem name not equal "BLSR"

Explanation: This message indicates that the subsystem name has been syntactically misspelled.

Action: Check for missing/extra commas, parens, quotes, etc.

#### DSS2252 Invalid BLSR subparm ""

Explanation: The indicated value is not a valid subparm for the BLSR subsystem.

Action: Refer to the IBM Batch LSR (BLSR) Subsystem Guide for valid subparms and their values.

#### DSS2254 Unexpected/Misplaced quote found in subparm name

Explanation: A quote was found before a valid delimiter while looking for a valid BLSR subparm name.

Action: Quotes must be coded at the beginning of the name.

#### DSS2256 Unexpected/Misplaced open paren found before subparm

Explanation: An open paren was found prior to a valid subparm name.

Action: Move the parenthesis after the name - enclose only the value in parentheses.

#### DSS2258 Misplaced close paren

Explanation: A close parenthesis was found somewhere other than at the end of a BLSR subparm value.

Action: Contact ASG Customer Support.

#### DSS2260 Subparm must be delimited by equal or open paren

Explanation: A BLSR subparm name must be followed by an equal or open paren only.

Action: Contact ASG Customer Support.

#### DSS2262 Opening quote must be preceded by a comma

Explanation: A complete BLSR subparm name and value must both be enclosed within quotes as a set.

Action: Multiple sets can be enclosed within one set of balanced quotes.

**DSS2264** Blanks only valid inside quoted strings

Explanation: This error is most likely caused by another error that caused the parsing processing to become lost.

Action: Correct all other errors and rescan JCL.

**DSS2266** Invalid delimiter "=" following subparm value

Explanation: This error caused when a subparm name is followed by 2 or more equals before a comma: parmname=value1=value2.

Action: Correct the coding error and rescan the JCL.

**DSS2268** Invalid delimiter "(" following subparm value

Explanation: This error caused when a subparm name is followed by an open paren before a comma: parmname=value1(value2).

Action: Correct the coding error and rescan the JCL.

**DSS2270** No valid delimiters found within subsys parm

Explanation: This error occurs when there are no equals, commas, parens, quotes, or blanks contained within the subsys string after the subsys name of BLSR.

Action: Contact ASG Customer Support.

**DSS2272** Missing delimiter after subparm value

Explanation: The end of the subsys parm was reached before a closing paren was found.

Action: The subsys parm length is limited to 240 characters.

**DSS2274** DDNAME "" is too long - 8 characters max

Explanation: The value for the BLSR subparm DDNAME was coded as more than the allowed maximum of 8 characters.

Action: This value must be a valid JCL DDNAME.

**DSS2276** First character of DDNAME can not be numeric

Explanation: Valid JCL DDNAMEs must start with an alpha character, \$, # or @ and must not be more than 8 characters in length.

Action: Contact ASG Customer Support.

**DSS2278** DDNAME "" contains invalid character ""

Explanation: The printed character is invalid within a JCL DDNAME.

Action: Only valid characters are alpha (caps), numeric and \$, #, @.

**DSS2280** Value "" for "" is too long

Explanation: The value specified for the noted subparm exceeds the maximum allowed length for that subparm.

Action: Refer to the IBM BLSR Subsystem Guide for allowed values for each valid subparm.

#### DSS2282 Value "" for "" is non-numeric

Explanation: The value specified for the noted subparm contains non-numeric characters.

Action: Refer to the IBM BLSR Subsystem Guide for allowed values for each valid subparm.

#### DSS2284 Value "" for "" is too low

Explanation: The value specified for the noted subparm is less than the allowed range.

Action: Refer to the IBM BLSR Subsystem Guide for allowed values for each valid subparm.

#### DSS2286 Value "" for "" is too high

Explanation: The value specified for the noted subparm is greater than the allowed range.

Action: Refer to the IBM BLSR Subsystem Guide for allowed values for each valid subparm.

#### DSS2288 Value "" for "" is not valid

Explanation: The value specified for the noted subparm is not one of the allowed values.

Action: Refer to the IBM BLSR Subsystem Guide for allowed values for each valid subparm.

#### DSS2290 Required subparm "DDNAME" not found

Explanation: The DDNAME subparm was not found within the subsys parm. It is a required value.

Action: Refer to the IBM BLSR Subsystem Guide for further information.

#### DSS2292 Unbalanced parens - excessive opens

Explanation: More left parens were found than right parens.

Action: There must be an equal number of left and right parens coded in the subsys parm.

#### DSS2294 Unbalanced parens - excessive closes

Explanation: More right parens were found than left parens.

Action: There must be an equal number of left and right parens coded in the subsys parm.

#### DSS2296 "" PARAMETER SPECIFIED MORE THAN ONCE

Explanation: Related IBM message - CSR003I

Action: The named parameter was specified more than once. This BLSR parameter can only be specified one time in the SUBSYS parm.

#### DSS2298E DD "" EXCEEDS 16 LSR POOL LIMIT

Explanation: Related IBM message - CSR004I

Action: The named DD statement requested that the subsystem select an unused SHRPOOL value for one or more batch local shared resources (LSR) requests. However, all 16 values (0-15) were already used.

#### DSS2300 JOBGROUP AND STARTBY ARE MUTUALLY EXCLUSIVE



Explanation: Both JOBGROUP and STARTBY parameters were found on a SCHEDULE statement. JOBGROUP and STARTBY are mutually Exclusive on a SCHEDULE statement.

Action: Remove either the JOBGROUP or STARTBY keywords form the STARTBY statement, as appropriate.

#### DSS2302 XXXXX IS INVALID PARAMETER FOR YYYYYY

Explanation: This message can be displayed for a number of reasons XXXXX is replaced by a parameter that was found to be invalid for YYYYYY, the command or statement in error.

Action: Correct the error as indicated by DSS2302.

#### DSS2304 NO MATCHING JOBGROUP FOR THIS XXXXX STATEMENT

Explanation: XXXXX is replaced by "ENDGROUP". This means an ENDCGROUP statement was found in a JOB, but a previous JOBGROUP statement was not found.

Action: Add a JOBGROUP statement prior to the ENDCGROUP statement.

#### DSS2305 MISSING REQUIRED NAME= FOR CONCURRENT STATEMENT

Explanation: A CONCURRENT statements was found in a JOB but the required NAME= parameter was either missing or misspelled.

Action: Either add a NAME= keyword to the CONCURRENT statement, Or, if misspelled, correct the spelling.

#### DSS2306 MAXIMUM NUMBER OF NAMES ON XXXXX

Explanation: XXXXX is replaced by either BEFORE or AFTER. For either, More than 10 names (NAME=(N1,N2...Nn), where n > 10) were specified on the required NAME= keyword.

Action: Reduce the number of names on the NAME= keyword to ten or less.

#### DSS2307 JOBNAME TOO LONG FOR NAME. EIGHT CHARS MAXIMUM

Explanation: A job name on a SCHEDULE statement was found to be longer than eight characters.

Action: Correct the jobname(s) on the AFTER or BEFORE keywords.

#### DSS2308 INCORRECT ABEND CODE. MUST START WITH "U" OR "S"

Explanation: On a BEFORE or AFTER statement, an ABENDCC was found but the value of ABENDCC was incorrect.

Action: Make sure the value of ABENDCC (i.e. ABENDCC=S0C4) Is a valid abend code. Abend codes must begin with "S" for system abends, or "U" for user abends.

#### DSS2309 NO MATCHING JOBSET NAME FOR THIS XXXXX

Explanation: XXXXX is replaced by ENDSET. An ENDSET statement was found but no matching JOBSET statement was before this ENDSET.

Action: Ensure there is a matching JOBSET statement prior to this ENDSET statement.

#### DSS2310 NO PRIOR XXXXX FOR THIS YYYYYY

Explanation: XXXXX is replaced by either JOBSET or JOBGROUP while YYYYY is replaced by either JOBGROUP or ENDGROUP.

Action: Supply the missing JOBSET or JOBGROUP statements.

#### DSS2311 DUPLICATE JOB NAME ON PRIOR JOBSET: XXXXX

Explanation: A JOBSET statement was found with the same job name as on a previous JOBSET statement.

Action: Ensure all jobnames within are unique.

#### DSS2312 XXXXX BUT NO MATCHING YYYYY

Explanation: Either a JOBSET or JOBGROUP statements was found but no matching ENDSET or ENDGROUP statements.

Action: Add appropriate ENDSET or ENDGROUP statements to complete the structure.

#### DSS2314 NO JOBNAME ON THIS SJOB STATEMENT

Explanation: An SJOB statement was found but a jobname was missing SJOB requires a jobname as a label as in “//Jobname SJOB”.

Action: Supply a required Jobname as a label on the SJOB statement.

#### DSS2315 SJOB OUTSIDE JOBSET CONTEXT

Explanation: An SJOB statement was specified but no prior JOBSET was found.

Action: Specify a JOBSET statement prior to specifying SJOB.

#### DSS2316 INCORRECT VALUE FOR XXXXX

Explanation: XXXXX is replaced by the “RC”. The value of RC on a WHEN or BEFORE statement is not numeric.

Action: Ensure the value of RC is numeric.

#### DSS2317 MISSING JOBNAME ON XXXXX STATEMENT

Explanation: A BEFORE or AFTER statements was found but a corresponding Job name was not found.

Action: Supply a job name on the BEFORE or AFTER statement.

#### DSS2319 MISSING JOB NAME FOR THIS JOBGROUP

Explanation: A SCHEDULE JOBGROUP statements was found, but the JOBGROUP Keyword was missing a job group name.

Action: Supply the missing job name as a value for the JOBGROUP keyword.

#### DSS2320 INVALID DATE FORMAT

Explanation: While validating the value of a DATE keyword, the format of the value was found to be incorrect.

Action: Correct the format of the value of the DATE keyword.

**DSS2321 XXXXX IS NOT UNIQUE FOR JOBGROUP**

Explanation: A JOBGROUP statements was found that had the same group name as a previous JOBGROUP statement.

Action: Ensure all group names are unique for all JOBGROP statements.

**DSS2322 MISSING NAME FOR THIS GJOB**

Explanation: Every GJOB statement requires a name. For example, “//GNAME GJOB ...”. A GJOB statement was found that has no name.

Action: Supply a unique name for this GJOB statement.

**DSS2323 MISSING GJOB BEFORE XXXXX**

Explanation: A BEFORE or AFTER statement was found but no prior GJOB statement was found.

Action: Supply the missing GJOB statement before the BEFORE or AFTER statements.

**DSS2324 ACCOUNTING INFO TOO LONG. 143 CHARS MAXIMUM**

Explanation: The accounting information on a JOB card was found to be more than 143 characters long.

Action: The maximum length of accounting information is 143 characters.

**DSS2326 UKNOWN OPERAND DETECTED**

Explanation: An unknown keyword was found on a JOBGROUP statements.

Action: Correct the keyword in error.

**DSS2328 INVALID SYNTAX FOR REGIONX**

Explanation: A syntax error was found on a REGIONX.

Action: Refer to MVS JCL Reference V2R2 or above for a description of the REGIONX parameter.

**DSS2329 REGION AND REGIONX ARE MUTUALLY EXCLUSIVE**

Explanation: On either a JOB or EXEC statement, both a REGIONX and REGION parameter were found. REGIONX and REGION are mutually exclusive.

Action: Remove either the REGIONX or REGION parameter, whichever is appropriate.

**DSS2330 INVALID NUMERIC VALUE FOR REGIONX**

Explanation: A REGIONX parameter contained an incorrect value that should be numeric.

Action: Correct the incorrect numeric as described in MVS JCL Reference V2R2 or greater.

**DSS2410 CONSTRUCT "name" CONSTRUCT NOT FOUND**

Explanation: Related IBM message - IGD01011I, IGD01014I, IGD17201I

Action: The construct (DATACLASS, STORCLAS, or MGMTCLAS) named in the JCL or returned by an ACS routine or ACS installation exit does not exist in the Active Storage Management subsystem configuration.

**DSS2412 VOLUME "" IS AN SMS-MANAGED VOLUME**

Explanation: Related IBM message - IGD301I

Action: A storage class was not derived or allowed for this dataset, for which an SMS-managed volume was explicitly specified. Only SMS-managed datasets may reside on SMS-managed volumes.

**DSS2414 VOLUME "" IS NOT AN SMS-MANAGED VOLUME**

Explanation: Related IBM message - IGD302I

Action: A storage class was derived for this dataset for which a non-SMS-managed volume was explicitly specified. SMS datasets can only reside on SMS-managed volumes. If the storage class is defined such that explicitly specified volumes are to be honored, this allocation will fail; otherwise, the dataset will be allocated on an SMS-managed volume.

**DSS2416 INCOMPATIBLE STORCLASS xxxxxxxx AND VOLUME yyyyyy**

Explanation: Related IBM message - IGD302I

Action: Storage Class xxxxxxxx was derived for this dataset and explicitly specified volumes are to be honored. Volume yyyyyy cannot be honored because it does not belong to a Storage Group that was selected for this dataset allocation.

**DSS2420 HSC/DFSMS CHANGED UNIT= FROM xxx TO yyy**

Explanation: Related STK message - SLS2373I

Action: The HSC/DFSMS Interface has changed the UNIT= value coded on the JCL DD statement (xxx) to the value yyy. This is an informational message.

**DSS2424 "text"**

Explanation: Related IBM message - IGD01004I

Action: The data class installation exit issued the message "text".

**DSS2425 "text"**

Explanation: Related IBM message - IGD01005I

Action: The storage class installation exit issued the message "text".

**DSS2426 "text"**

Explanation: Related IBM message - IGD01006I

Action: The management class installation exit issued the message "text".

**DSS2427 "text"**

Explanation: Related IBM message - IGD01007I

Action: The data class ACS routine issued the message "text".

**DSS2428 "text"**

Explanation: Related IBM message - IGD01008I

Action: The storage class ACS routine issued the message "text".

#### DSS2429 "text"

Explanation: Related IBM message - IGD01009I

Action: The management class ACS routine issued the message "text".

#### DSS2430 "text"

Explanation: Related IBM message - IGD01010I

Action: The storage group ACS routine issued the message "text".

#### DSS2431 LABEL ERROR AND/OR NON-LIBRARY VOLUME SPECIFIED

Explanation: Related IBM message - IGD01020I

Action: Either NSL or LTM was specified as a LABEL keyword and/or a non-library volume was specified when tape type storage group was selected.

#### DSS2432 ALLOCATION FAILED BY ACS ROUTINE/INSTALLATION EXIT

Explanation: Related IBM message - IGD01001I

Action: Either an ACS routine or ACS installation exit failed the data set allocation. This will be followed by message number DSS2433 containing the return and reason codes from the ACS routine or installation exit.

#### DSS2433 ACS EXECUTOR RETURNED CODE rc - REASON CODE rsnc

Explanation: Related IBM message - IGD01001I

Action: Either an ACS routine or ACS installation exit failed the data set allocation with a return code of "rc" and a reason code of "rsnc". This message follows message DSS2432.

#### DSS2510 DATA SET NOT ELIGIBLE FOR ALLOC ON SMS-MANAGED VOLUME

Explanation: Related IBM message - IGD301I

Action: A storage class was not derived or allowed for this dataset, for which an SMS-managed volume was explicitly specified. Only SMS-managed datasets may reside on SMS-managed volumes.

#### DSS2512 INCOMPATIBLE STORAGE CLASS AND VOLUME

Explanation: Related IBM message - IGD302I

Action: A storage class was derived for this dataset and an explicit volume request was specified. The volume request cannot be met for one of the following reasons:

- not all of the volumes are SMS-managed
- not all of the volumes are defined to the same storage group
- the storage group containing the volumes was not selected for this dataset allocation

#### DSS2516 ERROR DURING SMS PROCESSING

Explanation: Related IBM message - IGD306I

Action: An unexpected error occurred during Storage Management Subsystem processing. An error record has been written to SYS1.LOGREC.

#### DSS2518 ERROR IN AN ACS INSTALLATION EXIT

Explanation: Related IBM message - IGD307I

Action: The installation exit either terminated abnormally or returned an unknown return code.

#### DSS2520 MGMTCLAS OR STORCLAS SPECIFIED FOR NON-SMS DATA SET

Explanation: Related IBM message - IGD310I

Action: Storage class or management class was explicitly specified for a dataset which is a type that is not eligible to be SMS managed (for example: unmovable, ISAM, absolute track allocation).

#### DSS2525 SMS ASSIGNED NON-TEMPORARY OR VSAM DATA SET TO VIO

Explanation: Related IBM message - IGD01012I

Action: The storage group ACS routine assigned a non-temporary or VSAM dataset to a VIO storage group. This type of assignment is not allowed.

#### DSS2526 STORAGE GROUP NOT ASSIGNED BY SMS ACS ROUTINE

Explanation: Related IBM message - IGD01013I

Action: The storage class ACS routine assigned a storage class but the storage group ACS routine did not assign a storage group. This indicates a problem with the storage group ACS routine. Contact your storage administrator.

#### DSS2560 NO VOLUMES ASSOCIATED WITH STORAGE GROUP

Explanation: A storage group was assigned for this dataset; however, no volumes are defined in the configuration.

Action: Contact ASG Customer Support.

#### DSS2604 DATACLAS " " EXPLICITLY SPECIFIED

Explanation: Data class " " was explicitly specified for this allocation but was not derived or allowed by the SMS subsystem.

Action: This message will be accompanied by DSS2605 or DSS2606.

#### DSS2605 NO DATACLAS SELECTED BY ACS ROUTINE

Explanation: No data class was selected by the data class ACS routine to be used in this allocation, even though a data class name was explicitly specified (see message DSS2604).

Action: Contact ASG Customer Support.

#### DSS2606 DATACLAS " " SELECTED BY ACS ROUTINE

Explanation: Data class " " was selected by the data class ACS routine and will be used in this allocation even though a different data class name was explicitly specified (see message DSS2604).

Action: Contact ASG Customer Support.

#### DSS2614 STORCLAS "" EXPLICITLY SPECIFIED

Explanation: Storage class "" was explicitly specified for this allocation but was not derived or allowed by the SMS subsystem.

Action: This message will be accompanied by DSS2615 or DSS2616.

#### DSS2615 NO STORCLAS SELECTED BY ACS ROUTINE

Explanation: No storage class was selected by the storage class ACS routine to be used in this allocation even though a storage class name was explicitly specified (see message DSS2614).

Action: NOTE: This implies that the dataset is to be SMS-managed but without a storage class it will NOT be an SMS-managed dataset.

#### DSS2616 STORCLAS "" SELECTED BY ACS ROUTINE

Explanation: Storage class "" was selected by the storage class ACS routine and will be used in this allocation, even though a different storage class name was explicitly specified (see message DSS2614).

Action: Contact ASG Customer Support.

#### DSS2620 NON-SMS DATA SET REFERS TO SMS-MANAGED VOLUME

Explanation: Related IBM message - IGD304I

Action: The dataset in this DD statement is not allowed to be allocated in the storage group of the VOL=L=REF dataset.

#### DSS2621 SMS-MANAGED VOLUME SPECIFIED FOR NON-SMS DATA SET

Explanation: Related IBM message - IGD04004I

Action: The catalog entry indicates that the dataset is not SMS-managed, but an SMS managed volume serial was specified.

#### DSS2622 MGMTCLAS IGNORED FOR NON-SMS MANAGED DATA SET

Explanation: Related IBM message - IGD303I

Action: A storage class was not derived or allowed for this dataset; therefore, it is not SMS managed. Management class may not be explicitly specified for a non-SMS managed dataset.

#### DSS2624 MGMTCLAS "" EXPLICITLY SPECIFIED

Explanation: Management class "" was explicitly specified for this allocation but was not derived or allowed by the SMS subsystem.

Action: This message will be accompanied by DSS2625 or DSS2626.

#### DSS2625 NO MGMTCLAS SELECTED BY ACS ROUTINE

- Explanation: No management class was selected by the management class ACS routine to be used in this allocation, even though a management class name was explicitly specified (see message DSS2624).
- Action: Contact ASG Customer Support.

**DSS2626 MGMTCLAS " " SELECTED BY ACS ROUTINE**

- Explanation: Management class " " was selected by the management class ACS routine and will be used in this allocation, even though a different management class name was explicitly specified (see message DSS2624).
- Action: Contact ASG Customer Support.

**DSS2632 DATA SET NAMED BELOW NOT ELIGIBLE TO BE SMS-MANAGED**

- Explanation: Related IBM message - IEF648I, IGD316I
- Action: A dataset name enclosed within quotes (apostrophes) cannot be cataloged and therefore cannot be SMS-managed. A DSS8900 message containing the dataset name follows this message.

**DSS3008 SUBLIST WITHIN SUBLIST INCORRECT IN THE xxxx FIELD**

- Explanation: Related IBM message - IEF616
- Action: JOB/SCAN found a subparameter list that was specified within a subparameter list. This arrangement is incorrect in a JCL statement. Possibly, too many parentheses were used, so that a list appeared to be within a list.

**DSS3009 IMPROPER SUBPARAMETER LIST IN THE PARM FIELD**

- Explanation: An improper subparameter list was coded in the PARM field.
- Action: An example of an improper list is PARM=(ONE,(TWO,THREE),FOUR). (TWO,THREE) is incorrect.

**DSS3010 ADDRSPC on EXEC STATEMENT is not "VIRT" or "REAL"**

- Explanation: The ADDRSPC on the EXEC statement, if coded, must contain a value of "VIRT" or "REAL".
- Action: VIRT (usually the default) requests virtual storage and allows job or step to be paged. REAL requests central or real storage and the system cannot page.

**DSS3013 INCORRECT ACCOUNTING FIELD IN THE EXEC STATEMENT**

- Explanation: The ACCOUNTING field of the EXEC statement contains a syntax error.
- Action: Check for missing apostrophes, missing commas, etc. Consult the IBM JCL manual for the correct syntax of this parameter. (acct-#)(pa#,room,time,lines,cards,forms,copies,log,linect) Code a comma in place of each omitted parameter when other parameters follow. Max length is 142 characters. Subparameters containing special characters (blank,./)'(\*&=+) must be enclosed in single quotes. Apostrophes and ampersands must appear as double characters.

**DSS3017 LENGTH OF PARM FIELD EXCEEDS 100 CHARACTERS**

- Explanation: A PARM field exceeds the IBM maximum length of 100 characters.



Action: Contact ASG Customer Support.

#### DSS3020 MISSING EQUAL SIGN ON EXEC PARM

Explanation: The PARM keyword on the EXEC statement must be followed by an equal sign (=) and the value to be assigned.

Action: If there are any delimiters (such as a comma) embedded in the value, then the entire string must be enclosed in single quotes. //STEP3 EXEC PGM=PAY12,PARM='P1,345,P2=2'

#### DSS3025 ILLEGAL USE OF PARENS IN EXEC STATEMENT

Explanation: Parentheses are not required for a keyword on the EXEC statement.

Action: For example, the REGION parameter on the EXEC statement should not use parentheses.

This is NOT correct

```
//STEP1 EXEC PGM=IEFBR14, (REGION=40K)
```

This is correct

```
//STEP1 EXEC PGM=IEFBR14,REGION=40K
```

#### DSS3030 DYNAMNBR MUST CONTAIN NUMERIC VALUE

Explanation: Related IBM message - IEF674I

Action: Only numeric values may be coded in this parameter. Limits for 32K TIOT (task input output table) are 0-1635 minus the number of DD statements in the step.  
Limits for 64K TIOT (task input output table) are 0-3273 minus the number of DD statements in the step.

#### DSS3040 INVALID DELIMITER IN EXEC STATEMENT

Explanation: An invalid character was found after the EXEC verb.

Action: For example: //STEP1 EXEC)

#### DSS3041 INVALID USE OF PERIOD IN EXEC STATEMENT

Explanation: An invalid use of the period was found in an EXEC parameter.

Action: This could be caused by reversing the parameter.procstepname entry. For example:

```
//STEP1 EXEC PROC1,PS2.COND=(0,NE),
```

#### DSS3050 FIRST EXEC KEYWORD IS NOT "PROC" OR "PGM"

Explanation: Related IBM message - IEF630I or IEFC630I

Action: On the EXEC statement, the first keyword after the word EXEC must be either "PGM=" or "PROC=" or the name of a procedure to be executed.

#### DSS3060 EXEC KEYWORD DID NOT END WITH EQUAL SIGN

Explanation: On the EXEC statement, all keywords following the word EXEC must end with an equal sign (=) followed by the value of the keyword.

Action: Contact ASG Customer Support.

**DSS3070 INVALID PARAMETER ON EXEC STATEMENT**

- Explanation:** The EXEC statement contained a parameter that was not considered a valid parameter.
- Action:** This may be caused by a missing delimiter, space, quotes etc. Valid keywords that are not standard IBM keywords may be listed in Control Table AY to avoid DSS3070 errors. Table AY entries direct DOCU/TEXT or JOB/SCAN to accept your non-standard EXEC statement keywords instead of generating an error message.

**DSS3080 INVALID REGION PARAMETER FORMAT**

- Explanation:** The correct format for the REGION parameter is: REGION=nnnK, nnn is 1 thru 7 decimal numbers, from 1 thru 2096128 'K' refers to kilobytes (1024 bytes) REGION=nM, n is 1 thru 4 decimal numbers, from 1 thru 2047 'M' refers to megabytes (1024 kilobytes)
- Action:** Contact ASG Customer Support.

**DSS3090 INVALID COND PARAMETER FORMAT**

- Explanation:** The COND parameter must be in one of the following formats. COND=(code,operator,) or COND=((code,operator,step),(code,operator,step),...)
- Action:** "code" specifies a number which the system compares to the return code from the JOB step. "operator" specifies the type of comparison to be made to the return code. "step" specifies the exact JOB step that is to be considered in the test. If no JOB step name is specified, all previous steps are used in the tests. "step" is not allowed on JOB CARD COND parms.

**DSS3091 INVALID JOB COND PARAMETER FORMAT**

- Explanation:** Related IBM message - IEF632I
- Action:** The JOB COND parameter must be in the following format. COND=(code,operator) "code" specifies a number which the system compares to the return code from the JOB step. The Value may be from 0 through 4095. "operator" specifies the type of comparison to be made to the return code. Permissible operators are GT, GE, EQ, NE, LT and LE. STEP is not permissible on a JOB COND= parameter.

**DSS3100 INVALID TIME PARAMETER FORMAT**

- Explanation:** The TIME parameter of the JOB statement must be a numeric value not greater than 1440 in one of the following formats. TIME=(min,sec)  
TIME=min  
TIME=(,sec)
- Action:** For example, TIME=(1,30); TIME=5; TIME=(,30); TIME=1440

**DSS3110 RETURN CODE "" IN COND IS NOT VALID**

- Explanation:** The return code must be a numeric value between 0 and 4095.
- Action:** Contact ASG Customer Support.

**DSS3120 OPERATOR "" IN COND IS INVALID**

Explanation: The operator in the COND parameter must be one of the following: GT, GE, EQ, LT, LE, NE  
 Action: Contact ASG Customer Support.

#### DSS3130 MORE THAN 8 COND SUBPARAMETERS NOT ALLOWED

Explanation: Too many sub-parameters in the COND parameter.  
 Action: This could be caused by a missing right parenthesis. Example of error:  
 //STEP5 EXEC PROC=PAYROLL,COND.WEEK=((52,LT),ONLY

#### DSS3140 STEP "" NOT FOUND IN JOB

Explanation: The step name (or stepname.procstepname) in the COND parameter could not be found in the JOB.  
 Action: This is probably due to misspelling the step name or removing the step from the job without changing the COND parameter.

#### DSS3141 statement named "" is a duplicate

Explanation: The stepname portion of stepname.procstepname in the COND parameter or IF statement is a duplicate and may not be resolved.  
 Action: The stepname refers to the EXEC statement that calls the cataloged or instream procedure and the procstepname refers to the EXEC PGM= statement within that procedure.

#### DSS3145 COND PARM ERROR. TEST OF CURRENT STEP NOT ALLOWED

Explanation: The step name (or stepname.procstepname) in the COND parameter cannot refer to the same step it is coded on.  
 Action: Contact ASG Customer Support.

#### DSS3150 INVALID CHARACTER IN PARM FIELD

Explanation: The Parm field contained an invalid character.  
 Action: Contact ASG Customer Support.

#### DSS3160 REGION MUST END IN A SINGLE CHARACTER "K" OR "M"

Explanation: The correct format for the REGION parameter is: REGION=nnnK,  
 where  
 nnn is 1 thru 7 decimal numbers, from 1 thru 2096128  
 'K' refers to kilobytes (1024 bytes)  
 REGION=nM,  
 where  
 n is 1 through 4 decimal numbers, from 1 through 2047  
 'M' refers to megabytes (1024 kilobytes)  
 Action: Contact ASG Customer Support.

#### DSS3170 VALUE OF REGION NOT NUMERIC

Explanation: The correct format for the REGION parameter is: REGION=nnnK,  
 where:

nnn is 1 thru 7 decimal numbers, from 1 thru 2096128  
 'K' refers to kilobytes (1024 bytes)  
 REGION=nM,  
 where:  
 n is 1 thru 4 decimal numbers, from 1 thru 2047  
 'M' refers to megabytes (1024 kilobytes)

Action: Contact ASG Customer Support.

#### DSS3171 REGION SIZE VALUE EXCEEDS 2096128K

Explanation: The REGION size value specified exceeds the maximum allowable size of 2096128 kilobytes.

Action: Contact ASG Customer Support.

#### DSS3172 REGION SIZE VALUE EXCEEDS 2047M

Explanation: The REGION size value specified exceeds the maximum allowable size of 2047 megabytes.

Action: Contact ASG Customer Support.

#### DSS3180 PROGRAM "" MARKED AS LOAD ONLY

Explanation: The program specified has been marked by the Linkage Editor as "load only".

Action: Check the output from the Link Edit job before rerunning.

#### DSS3190 MODULE "" MARKED NOT EXECUTABLE

Explanation: The program specified has been marked by the Linkage Editor as "not executable".

Action: Check the output from the Link Edit job before rerunning.

#### DSS3200 INVALID DYNAMNBR

Explanation: Related IBM message - IEF674I

Action: Only numeric values may be coded in this parameter. Limits for 32K TIOT (task input output table) are 0-1635 minus the number of DD statements in the step.  
 Limits for 64K TIOT (task input output table) are 0-3273 minus the number of DD statements in the step.

#### DSS3210 EXTRANEIOUS VALUE "" IN COND

Explanation: The COND parameter must be in one of the following formats. COND=(code,operator,) or COND=((code,operator,step),(code,operator,step),...) or COND=EVEN or COND=ONLY

Action: "code" specifies a number which the system compares to the return code from the JOB step.  
 "operator" specifies the type of comparison to be made to the return code.

#### DSS3220 PROGRAM PARM REQUIRED AND NOT FOUND

Explanation: Control Table AP indicates that the program name will be found in the Parm field, but no Parm field was found. (IMS programs fall into this category.) Control Table AP specifies program

names for which substitution applies. DOCU/TEXT or JOB/SCAN checks each program against Table AP.

Action: If a program name from an EXEC statement matches a program name in Table AP, the true program is extracted from the PARM list.

#### DSS3230 "" COND CODE MATCH FOR: JOBSTEP PROCSTEP

Explanation: Information message from JOB/SCAN when CONDCODE processing is turned on saying that a 'JOB' or 'PROC' or 'EXEC' condition code parameter was found which will cause this step not to be executed.

Action: The PROCESSING SPECIFICATIONS (PSS) Run Option statement controls the effect of COND parameters during the DOCU/TEXT or JOB/SCAN process. If PSS=CONDCODE is coded, Control Table AR is checked. Information included in Table AR is used by DOCU/TEXT or JOB/SCAN to assign a return code other than zero or an abend to a jobstep, procstep or program step. The Table's return code is applied as if the step had actually ended with that return code. This step and all following are flushed and only syntax checking is performed.

#### DSS3231 MATCHING COND CODE XXXXX YY JOBSTEP PROCSTEP

Explanation: Information message from JOB/SCAN giving the cond parameter that caused this step not to be executed.

Action: XXXXX is 0 --> 4095 or 'ABEND'.  
YY is the operator - (ie: EQ or NE or GT or LT or GE or LE). (see also Message DSS3230)

#### DSS3235 PROCESSING BYPASSED DUE TO IF-THEN CONDITION

Explanation: Information message from JOB/SCAN when condcod processing is turned on saying that an IF-THEN condition was false which will cause this step not to be executed.

Action: Contact ASG Customer Support.

#### DSS3240 REMAINDER OF JOB FLUSHED DUE TO "JOB" COND CODE

Explanation: Information message from JOB/SCAN saying that a cond parameter on the JOB card matched the return code from a prior step.

Action: This step and all that follow are flushed by JOB/SCAN - only syntax checking is done. (See also Message DSS3230)

#### DSS3250 JOBSTEP PROCSTEP RETURN CODE IS XXXXX ("AR" TABLE)

Explanation: Information message from JOB/SCAN saying that this step was entered in Table AR and the simulated COND CODE was XXXXX which can be any numeric value from 0 --> 4095 or 'ABEND'. (See also Message DSS3230)

Action: Contact ASG Customer Support.

#### DSS3300 JOB CONTAINS MORE THAN 255 STEPS (MVS LIMIT)

Explanation: A job can have a maximum of 255 steps. This maximum includes all steps in any procedures the EXEC statements call.

Action: Contact ASG Customer Support.

#### DSS3400 INVALID FORMAT FOR INCLUDE STATEMENT

Explanation: Format for INCLUDE statement is incorrect.

Action: Should be //name INCLUDE MEMBER=mbrname.

#### DSS3410 "" IS INVALID WITHIN INCLUDE MEMBER

Explanation: JOB, PROC, PEND, JCLLIB, JES2/JES3, and JES3 DD \* and DD DATA statements are not valid in an INCLUDE member.

Action: Contact ASG Customer Support.

#### DSS3420 INCLUDE MBR "" NOT FOUND

Explanation: The INCLUDE member specified could not be found in any of the libraries available to JOB/SCAN.

Action: Check the name of the INCLUDE statement for spelling errors and make sure the proper libraries are included in the set available to JOB/SCAN.

#### DSS3422 RECURSIVE INCLUDE OF MEMBER membername

Explanation: Processing of this INCLUDE statement stops. This INCLUDE statement includes itself indefinitely.

Action: Contact ASG Customer Support.

#### DSS3500 UNIDENTIFIED XMIT PARAMETER ""

Explanation: The XMIT statement contains a parameter or a keyword that is not identifiable.

Action: Possible causes are misspelling, extraneous space, missing comma, etc.

#### DSS4000 Volume count exceeds limit

Explanation: Related IBM message - IEF638I

Action: The maximum volume count for a VSAM or SMS-managed dataset is 59, otherwise the maximum volume count is 255.

#### DSS4010 VOLUME SERIAL NUMBER EXCEEDS 6 CHARACTERS

Explanation: The value for the VOL=SER parameter must be 1 to 6 characters in length.

Action: Up to 255 serial numbers may be coded, separated by a comma. A missing comma may cause this error if you have multiple serial numbers coded.

#### DSS4012 SERIAL VALUE MUST BE IN QUOTES - SPECIAL CHARACTERS.

Explanation: The value for the VOL=SER parameter must be enclosed in quotes if it contains special characters.

Action: The special characters are:  
 , - Comma  
 . - Period  
 / - Slash

' - Quote

( - Left Paren

) - Right Paren

\* - Asterisk

& - Ampersand

+ - Plus sign

= - Equal

All are valid within the volume serial parameter if the entire six character serial number is enclosed in single quotes.

#### DSS4013 SERIAL VALUE CONTAINS INVALID CHARACTERS

**Explanation:** The value for the VOL=SER parameter may contain Alphanumeric, National (\$,#,@), or Special characters.

**Action:** If special characters are used, the six character volume serial value should be enclosed in apostrophes.

#### DSS4014 SERIAL VALUE CONTAINS ALL BLANKS

**Explanation:** The value for the VOL=SER parameter may contain Alphanumeric, National (\$,#,@), or Special characters.

**Action:** This may be caused by coding two commas in a row in the SER field for multiple serial numbers.

#### DSS4030 THE BURST PARAMETER HAS AN IMPROPER VALUE

**Explanation:** Related IBM message - IEF643I

**Action:** The BURST parameter, when coded, must contain one of the following values: - 'N' - 'Y' - 'NO' - 'YES' The default value is 'N'.

#### DSS4040 VALUE OF CHKPT MUST BE "EOV"

**Explanation:** The CHKPT parameter, when coded, must contain the value "EOV". CHKPT=EOV

**Action:** Contact ASG Customer Support.

#### DSS4050 DATA SETNOT FOUND IN CATALOG

**Explanation:** Related IBM messages - IEF283I, IEF286I

**Action:** The dataset specified in the DD statement with a DISP of OLD or SHR could not be found in the catalog. This could be due to a misspelled DSN or to the fact that the dataset may have been uncataloged previously in the set of jobs being scanned. This message will also be issued if the SER subparameter of the VOLUME parameter specified an SMS-managed volume and the dataset was not cataloged.

#### DSS4052 CATALOG INDICATES DATA SET MIGRATED/ARCHIVED

**Explanation:** The dataset specified in the DD statement with a DISP of OLD or SHR was found in the catalog; however, the volume serial indicates that the dataset has been migrated (archived) to another device.

Action: Contact ASG Customer Support.

#### DSS4055 DATA SET NOT FOUND. UNPREDICTABLE RESULTS POSSIBLE

Explanation: Related IBM message - IEF283I

Action: A dataset was not found in the catalog, but the disposition on the JCL indicated that one was expected to already exist. For example: DISP=(MOD,UNCATLG) If this job is run and the dataset does not exist, one will be created.

#### DSS4070 VALUE OF COPIES PARAMETER IS NOT NUMERIC

Explanation: Related IBM message - IEF681I

Action: The values coded in the COPIES parameter must be numeric and within the range of 1-255 for JES2 and 1-254 for JES3. MVS uses the DD COPIES default of 1 when this parameter is incorrectly coded.

#### DSS4075 VALUE OF KEYOFF PARAMETER IS NOT NUMERIC

Explanation: The values coded in the KEYOFF parameter must be numeric.

Action: KEYOFF=offset-to-key

#### DSS4076 VALUE OF "BLKSIZE" PARAMETER GREATER THAN maximum

Explanation: The value coded in the BLKSIZE parameter is greater than the maximum allowed.

Action: The maximum allowed value is 2147483648 (two gigabytes). This may be specified as 2097152K or 2048M or 2G or 2147483648.

#### DSS4077 VALUE OF BLKSIZE PARAMETER IS NOT NUMERIC

Explanation: The value coded in the BLKSIZE parameter is not an integer.

Action: This parameter may be coded value, valueK, valueM, or valueG, but "value" must be numeric.

#### DSS4078 INVALID FORMAT FOR BLKSIZE PARAMETER

Explanation: The correct format for the BLKSIZE parameter is: BLKSIZE=value specifies the maximum length, in bytes, of a block  
BLKSIZE=valueK specifies the maximum length, in kilobytes, of a block. (1 kilobyte = 1024 bytes)  
BLKSIZE=valueM specifies the maximum length, in megabytes, of a block. (1 megabyte = 1024 kilobytes).  
BLKSIZE=valueG specifies the maximum length, in gigabytes, of a block. (1 gigabyte = 1024 megabytes).

Action: Contact ASG Customer Support.

#### DSS4080 VALUE OF BLKSIZE SUBPARAMETER IS NOT NUMERIC

Explanation: The value coded for the BLKSIZE subparameter of the DCB parameter must be numeric and within the range of 18 through 32,760.

Action: Contact ASG Customer Support.



**DSS4081 "" SUBPARM IS MORE THAN "" DIGITS LONG**

Explanation: The number of digits coded exceeds the maximum number allowed for the specified sub-parameter.

Action: The number of digits must not exceed the specified digits allowed.

**DSS4082 VALUE OF "BLKSZLIM" PARAMETER GREATER THAN maximum**

Explanation: The value coded in the BLKSZLIM parameter is greater than the maximum allowed. The maximum allowed value is 2147483648 (two gigabytes).

Action: This may be specified as 2097152K or 2048M or 2G or 2147483648.

**DSS4083 VALUE OF "BLKSZLIM" PARAMETER LESS THAN minimum**

Explanation: The value coded in the BLKSZLIM parameter is less than the minimum allowed.

Action: The minimum allowed value is 32760 bytes or 32K or 1M or 1G depending on the format of the BLKSZLIM value.

**DSS4084 INVALID FORMAT FOR BLKSZLIM PARAMETER**

Explanation: The correct format for the BLKSZLIM parameter is: BLKSZLIM=value specifies in bytes an upper limit on a dataset's block size.  
BLKSZLIM=valueK specifies the block size limit in kilobytes (units of 1024).  
BLKSZLIM=valueM specifies the block size limit in megabytes (units of 1024K).  
BLKSZLIM=valueG specifies the block size limit in gigabytes (units of 1024M).

Action: Contact ASG Customer Support.

**DSS4085 VALUE OF BLKSZLIM PARAMETER IS NOT NUMERIC**

Explanation: The value coded in the BLKSZLIM parameter is not an integer.

Action: This parameter may be coded value, valueK, valueM, or valueG, but "value" must be numeric.

**DSS4087 BLKSIZE CANNOT EXCEED 32760 FOR DASD DATA SET**

Explanation: The value coded in the BLKSIZE parameter is greater than 32760, the maximum allowed for a dataset on a DASD device.

Action: Contact ASG Customer Support.

**DSS4088 BLKSIZE CANNOT EXCEED 256KB FOR TAPE DATA SET**

Explanation: The value coded in the BLKSIZE parameter is greater than 256KB (262144 bytes), the maximum allowed for a dataset on a TAPE device.

Action: Contact ASG Customer Support.

**DSS4090 VALUE OF LRECL SUBPARAMETER IS NOT NUMERIC**

Explanation: The value coded for the LRECL parameter must be numeric and it must not contradict the RECFM and BLKSIZE values.

Action: The only time a non-numeric value is allowed is LRECL=x for a QSAM file that has an LRECL larger than 32,760 and is variable length spanned.

#### DSS4095 SEGMENT PARAMETER NOT NUMERIC OR INVALID VALUE

Explanation: The value coded for the SEGMENT parameter of the OUTPUT statement must be numeric and in the range of 1 - 99999.

Action: Contact ASG Customer Support.

#### DSS4100 VALUE OF BUFNO SUBPARAMETER IS NOT NUMERIC

Explanation: The value coded for the BUFNO subparameter of the DCB parameter must be numeric.

Action: Contact ASG Customer Support.

#### DSS4105 VALUE OF BUFNO SUBPARAMETER IS OUT OF RANGE

Explanation: The value coded for the BUFNO subparameter of the DCB parameter must be between 1 and 255.

Action: Contact ASG Customer Support.

#### DSS4110 VALUE OF DEN SUBPARAMETER IS NOT NUMERIC

Explanation: The value of the DEN subparameter of the DCB parameter must be:  
0-4,  
where  
0 = 7-track 200 BPI,  
1 = 7-track 556 BPI,  
2 = 7-track 800 BPI or 9-track 800 BPI,  
3 = 9-track 1600 BPI,  
4 = 9-track 6250 BPI.

Action: Contact ASG Customer Support.

#### DSS4115 "OPTCD VALUE TOO LONG OR MISSING

Explanation: The value of the OPTCD subparameter of the DCB parameter contains too many characters or is missing.

Action: See the IBM JCL Reference Manual, DD statement, DCB parameter, OPTCD subparameter for valid values.

#### DSS4116 "OPTCD" VALUE IS NOT LEGAL

Explanation: The value of the OPTCD subparameter of the DCB parameter contains an invalid value.

Action: See the IBM JCL Reference Manual, DD statement, DCB parameter, OPTCD subparameter for valid values.

#### DSS4120 LENGTH OF DD-NAME GREATER THAN 8 CHARACTERS

Explanation: The length of a DD name cannot be longer than 8 characters.

Action: This can occur in DDNAME= referbacks.

#### DSS4125 VALUE OF "FILEDATA" PARM IS NOT "BINARY" OR "TEXT"

Explanation: The value coded for the FILEDATA parameter of the DD statement must be either "BINARY" or "TEXT".

Action: Contact ASG Customer Support.

#### DSS4130 VALUE OF FREE PARAMETER IS NOT "END" OR "CLOSE"

Explanation: Related IBM message - IEF682I

Action: The value coded for the FREE parameter of the DD statement must be either "END" or "CLOSE".

#### DSS4140 VALUE OF HOLD PARAMETER IS NOT "YES" OR "NO"

Explanation: Related IBM message - IEF684I

Action: The value of the HOLD parameter., if coded, must contain either "YES", or "NO".

#### DSS4150 INVALID SUBPARAMETER IN LABEL PARAMETER FIELD

Explanation: The LABEL parameter has the following format:

```

1 2 3 4 5
LABEL=(dataset-seq-num, label-type, PASSWORD, IN ,RETPD=nnnn)
, ,NOPWREAD,OUT,EXPDT=yyddd
,

```

Action: Note that the first 4 parameters are positional and a comma is substituted when one is left out. If all 4 are left out then no commas are needed before the fifth keyword parameter. These are the values:

dataset-seq-num, 1 thru 4 decimal numbers  
label-type, SL,SUL,AL,AUL,NSL,NL,BLP,LTM (default SL)  
nnnn, RETPD days, value 1 thru 4 decimal numbers  
yyddd, EXPDT date, value yy=01 to 99, ddd=001 to 366

#### DSS4160 INVALID LABEL PARAMETER FORMAT

Explanation: The LABEL parameter has the following format:

```

1 2 3 4 5
LABEL=(dataset-seq-num, label-type, PASSWORD, IN ,RETPD=nnnn)
, ,NOPWREAD,OUT,EXPDT=yyddd
,

```

Action: Note that the first 4 parameters are positional and a comma is substituted when one is left out. If all 4 are left out then no commas are needed before the fifth keyword parameter. These are the values:

dataset-seq-num, 1 thru 4 decimal numbers  
label-type, SL,SUL,AL,AUL,NSL,NL,BLP,LTM (default SL)  
nnnn, RETPD days, value 1 thru 4 decimal numbers  
yyddd, EXPDT date, value yy=01 to 99, ddd=001 to 366

#### DSS4165 LGSTREAM VALUE IS LONGER THAN 26 CHARACTERS

- Explanation: The name specified in the LGSTREAM parameter is longer than the maximum length of twenty-six (26) characters.
- Action: Contact ASG Customer Support.

#### DSS4166 LGSTREAM VALUE IS INVALID

- Explanation: The name specified in the LGSTREAM parameter is invalid.
- Action: The name, up to a maximum of twenty-six characters, consists of one or more segments. Each segment may contain one to eight characters, which may be alphabetic, numeric, or national (\$,#,@) characters. Segments are joined by periods with periods being counted as characters towards the limit of twenty-six. The first character of each segment must be non-numeric.

#### DSS4170 VALUE OF PROTECT PARAMETER IS NOT "YES"

- Explanation: The PROTECT parameter, if coded, must have a value of "YES" or "Y".
- Action: Contact ASG Customer Support.

#### DSS4174 VALUE OF EATTR PARAMETER IS NOT "OPT" OR "NO"

- Explanation: The EATTR parameter, if coded, must have a value of "OPS" or "NO".
- Action: Contact ASG Customer Support.

#### DSS4180 INVALID STRUCTURE FOR QNAME

- Explanation: The QNAME cannot be longer than 8 characters.
- Action: This may be caused by either a space between the QNAME and the next parameter or a spelling error that made the name too long. QNAME=procname

#### DSS4200 NON-NUMERIC VALUE FOR SPACE AMOUNT

- Explanation: The SPACE parameter must contain at least one numeric value in the quantity field of the parameter.
- Action: Verify that all the positional sub-parameters are accounted for and check for spelling errors.

#### DSS4205 NON-NUMERIC VALUE FOR " "

- Explanation: The value for the specified parameter must be numeric.
- Action: Contact ASG Customer Support.

#### DSS4210 COMMA MISSING IN UNIT PARM - FORMAT IS (XXXX,,DEFER)

- Explanation: The UNIT parameter, in this format, has three sub-parameters.
- Action: When the second sub-parameter is skipped, its position must be indicated with the second comma prior to the DEFER.

#### DSS4220 UNIT COUNT CONTAINS NON-NUMERIC VALUE, MUST BE 1-59

- Explanation: The unit count sub-parameters of the UNIT parameter, when coded, must contain a numeric value of 1-59.

Action: Contact ASG Customer Support.

#### DSS4230 THIRD VALUE OF UNIT PARAMETER IS NOT "DEFER"

Explanation: The only value allowed in the third sub-parameter of the UNIT parameter is "DEFER".

Action: This message usually is caused by a spelling error.

#### DSS4240 STATEMENT NAME CONTAINS INVALID CHARACTER

Explanation: Related IBM message - IEF647I, IEF662I or IEF662I

Action: The statement name must begin with an alphabetic character or one of the National Character Set (\$,#,@) and must not start with a number. The rest of the name may contain alphanumeric and National characters.

#### DSS4250 "" MUST PRECEDE ANY STEPS

Explanation: Related IBM message - IEF636I, IEF671I

Action: The JOBLIB and JOBCAT DD statements must be placed before any steps to be executed in the job stream. This could be caused by the addition of a new step at the beginning of the JOB.

#### DSS4251 "" CANNOT BE USED WITH SMS DATA SETS

Explanation: Related IBM messages - IGD309I, IGD04002I

Action: The JOBCAT and STEPCAT DD statements cannot be used with SMS-managed datasets. JOBCAT can only be used when no SMS-managed datasets are referenced in the job. STEPCAT can only be used when no SMS-managed datasets are referenced in the step.

#### DSS4252 JOBLIB DD MUST PRECEDE "" DD

Explanation: Related IBM message - IEF636I

Action: The SYSCHK DD (related to the RESTART processing) and the JOBCAT DD must follow the JOBLIB DD (if one is present). Either DD preceding the JOBLIB DD results in the above message.

#### DSS4253 "ABSTR" CANNOT BE USED WITH SMS DATA SETS

Explanation: The SPACE sub-parameter ABSTR (absolute track) is not valid when the dataset is being controlled by SMS.

Action: Contact ASG Customer Support.

#### DSS4254 JOBCAT OR STEPCAT NOT PERMITTED

Explanation: Related IBM message - IEF6034I

Action: A JOBCAT or STEPCAT was encountered in the current job but JOBCAT/STEPCAT statements are no longer permitted starting with z/OS V1R7. If you are running on z/OS V1R4, V1R5 or V1R6, JOBCAT/STEPCAT statements may be disabled. If your installation has issued the system MODIFY CATALOG command to enable JOBCAT/ STEPCAT statements, this message may be suppressed or the severity of the message may be changed using the Standards Facility or the 'AW' Control table.

**DSS4255 DDNAME "" CANNOT APPEAR BEFORE EXEC**

Explanation: A ddname was found before the first EXEC statement in a job and it is not JOBLIB or JOBCAT.  
Action: Contact ASG Customer Support.

**DSS4260 BLKSIZE IS NOT A MULTIPLE OF LRECL IN RECFM=FB**

Explanation: Related IBM message - IEC041I, ABEND S013-20 or S03B  
Action: When specifying the DCB for a fixed blocked dataset, the BLKSIZE value must be a multiple of the value entered for the LRECL.

**DSS4270 BLKSIZE "" IS NOT FIRST IN CONCATENATION**

Explanation: Related IBM message - IEC020, ABEND S001-1  
Action: When concatenating datasets, you must place the dataset with the largest BLKSIZE first or code a "BLKSIZE" parameter that specifies the largest BLKSIZE on the first DD statement. In the message, the value of the largest BLKSIZE is displayed so that you may code it. NOTE -- Certain MVS access methods allow concatenating datasets in any order regardless of BLKSIZE. If your installation uses these, you may want to suppress this message or downgrade it to a warning or an advisory. Refer to the IBM Using Data Sets manual for a more thorough discussion of dataset concatenation.

**DSS4280 "" INVALID / MISPLACED IN SPACE PARM**

Explanation: Related IBM message - IEF640I or IEF640I  
Action: The value displayed is placed incorrectly in the SPACE parameter. Verify the format and correct.

**DSS4290 STATEMENT NAME LONGER THAN 8 CHARACTERS**

Explanation: Related IBM message - IEF642I  
Action: The name of a statement (immediately following the "//") must not be longer than 8 characters. Check for spelling error or missing space following name.

**DSS4310 DCB KEYWORD DID NOT END WITH EQUAL SIGN**

Explanation: A keyword in the DCB string was not followed by an equal sign (=).  
Action: The format is KEYWORD=value.

**DSS4330 MISSING DDNAME VALUE**

Explanation: A DDNAME= referback is missing the DD name value which is necessary for proper identification of the DD statement to be overridden.  
Action: The format is //STEPNAME.DDNAME DD ----

**DSS4340 INVALID VALUE FOR TERM; MUST BE "TS"**

Explanation: The TERM parameter, when coded, must be assigned a value of "TS". TERM=TS  
Action: Contact ASG Customer Support.

**DSS4350 UNIDENTIFIED VALUE FOR DISP**

- Explanation: The following are the values allowed in the first DISP sub-parameter: NEW, OLD, SHR, SHARE, MOD, or ', ' following the left parenthesis.
- Action: The second sub-parameter can contain PASS, KEEP, CATLG, UNCATLG, or DELETE. The third sub-parameter can contain DELETE, KEEP, CATLG, or UNCATLG.

**DSS4351 DISP NOT SPECIFIED - DISP=(NEW,DELETE) ASSUMED**

- Explanation: This DD statement contains a dataset name that is already cataloged and the DISP parameter was omitted and defaults to DISP=(NEW,DELETE).
- Action: Contact ASG Customer Support.

**DSS4352 DISP FIELD INCOMPATIBLE WITH DSNAME**

- Explanation: Related IBM message - IEF286I.
- Action: A GDG dataset was specified as "NEW" but currently exists. A GDG dataset was specified as "OLD" or "SHR" but the dataset does not exist.

**DSS4355 UNKNOWN POSITIONAL VOLUME SUBPARAMETER**

- Explanation: A subparameter in the VOL= string is unrecognized.
- Action: This can be caused by a misspelled subparameter or an incorrect number of commas.

**DSS4360 UNKNOWN VOL SUBPARAMETER**

- Explanation: The valid subparameters for the VOL parameter are SER, REF, or PRIVATE.
- Action: This error can be caused by a misspelled or missing subparameter.

**DSS4365 UNKNOWN LABEL SUBPARAMETER**

- Explanation: A LABEL subparameter is unrecognized.
- Action: This can be caused by non-numeric data or a misplaced or misspelled value in the LABEL parameter.

**DSS4367 UNKNOWN OR MISSING " " VALUE**

- Explanation: The value specified for this parameter is not valid.
- Action: See the IBM JCL Reference Manual for valid values.

**DSS4370 UNIDENTIFIED DD PARAMETER**

- Explanation: Related IBM message - IEF630I or IEFC630I
- Action: The DD statement contains a parameter or a keyword that is not identifiable. Possible causes are misspelling, extraneous space, missing comma, etc.

**DSS4380 DUPLICATE DDNAME " "**

- Explanation: Related IBM message - IEF654I,IEF672,IAT4410

Action: Two or more DD statements with the same ddname occurred within the same step. The second occurrence will be ignored if left uncorrected in a JES2 environment. The job will be cancelled in a JES3 environment.

#### DSS4390 NO UNIT SPECIFIED FOR NEW DATA SET

Explanation: When creating a new dataset, you must specify the type of UNIT it is to reside on.

Action: This error could be caused by a misspelling of the keyword UNIT as INIT. In a case like this, this message is usually accompanied by message DSS4370.

#### DSS4400 SPACE NOT SPECIFIED FOR NEW DASD DATA SET

Explanation: Related IBM message - IEF127I, IGD17045I

Action: When creating a new DASD dataset, you must specify the amount of space to be allocated. This may be done either explicitly with the SPACE JCL DD parameter or implicitly with a LIKE, REFDD, or DATACLAS DD parameter. This error could be caused by a misspelled keyword (usually accompanied by message DSS4370).

#### DSS4410 RELATIVE GENERATION ALREADY EXISTS

Explanation: An attempt was made to create a relative generation that already exists.

Action: Contact ASG Customer Support.

#### DSS4420 GENERATION DATA GROUP DOES NOT EXIST

Explanation: The generation dataset specified on the DD statement does not have a corresponding GDG in the system catalog.

Action: Either the BLDG (IEHPROGM) or DEFINE GENERATIONDATAGROUP (IDCAMS) was not done prior to this step or the dataset name was misspelled.

#### DSS4450 TEMPORARY DATA SET PREVIOUSLY DEFINED

Explanation: A temporary DSN was coded twice in the same job for output.

Action: When it is later used as input, only the first occurrence will be used.

#### DSS4460 GENERATION DOES NOT EXIST

Explanation: The GDG, referenced by absolute or relative generation, does not exist or has not been cataloged.

Action: Contact ASG Customer Support.

#### DSS4465 INVALID DISP ON TEMP DATA SET, "PASS" WILL BE USED

Explanation: For a new temporary dataset, the second or third DISP parameter may not be KEEP or CATLG.

Action: This is because a temporary dataset is always deleted at the end of the job.

#### DSS4466 "CATLG" INVALID DISPOSITION, "KEEP" WILL BE USED

Explanation: Related IBM message - IEF648I, IGD316I



Action: A dataset name enclosed within quotes (apostrophes) cannot be cataloged. The disposition is changed to "KEEP" by the operating system.

#### DSS4470 "CATLG" INVALID ON TEMP DATA SET, "PASS" WILL BE USED

Explanation: For a new temporary dataset, the second or third DISP parameter may not be KEEP or CATLG.

Action: This is because a temporary dataset is always deleted at the end of the job.

#### DSS4480 "KEEP" INVALID ON TEMP DATA SET, "PASS" WILL BE USED

Explanation: Related IBM message - IEF648I

Action: For a new temporary dataset, the coded or implied second or third DISP parameter may not be KEEP or CATLG. This is because a temporary dataset is always deleted at the end of the job. If an implied DISP will be KEEP or CATLG (ex:DISP=MOD), you must override the implied DISP with a PASS or DELETE.

#### DSS4490 UNIT AFF INVALID ON NEW DASD DATA SET

Explanation: UNIT=AFF=ddname may not be coded for a new DASD dataset. Remove the AFF parameter.

Action: If the dataset needs to be written to a specific volume, code the VOL=SER= parameter and specify the required volume serial number.

#### DSS4500 INVALID DATA SET NAME FORMAT IN " " OPERAND

Explanation: Related IBM message - IEF366I,IEF650I,IEF651I,IEF624I IEF6501,IEFC651I

Action: The DSN operand does not meet the requirements of temporary or permanent dataset naming conventions.

A temporary dataset name is one or two ampersands followed by one through nine alphanumeric or national characters, a hyphen, or a plus zero. The first character is alphabetic or national.

A permanent dataset is assigned either an unqualified or a qualified name.

An unqualified name is one through eight alphanumeric or national characters followed by a hyphen. The first character is alphabetic or national.

A qualified name consists of multiple names or nodes joined by periods. Each name or node is coded like an unqualified name of one through eight characters.

The maximum length of a qualified dataset name is 44 characters, including periods. For a GDG the maximum is 35 characters, including periods. For output tape the maximum is 17 characters, including periods.

The first character of a relative generation number must be +, -, or 0.

A relative generation number cannot exceed 255.

All characters of a relative generation number must be numeric (0-9).

A relative generation number must be expressed in 1-3 numeric characters. It may be prefaced by a + or -, for example +101, -002, +4, -09, 000.

With IMS, a member name for the IMS Proclib dataset is created from the PARM= fields 12 and 19 from the EXEC PGM=M=DFSRRRC00. This member name must also conform to MVS naming conventions.

**DSS4520 VOL=REF DATA SET NOT FOUND**

Explanation: Related IBM messages - IEF212I, IGD17110I

Action: The dataset name coded in the VOL=REF parameter was not found. The wrong dataset name was used or the name was misspelled.

**DSS4525 VOL=REF DATA SET HAS BEEN MIGRATED/ARCHIVED**

Explanation: The dataset name coded in the VOL=REF parameter was found in the catalog but the volume serial indicates that it has been migrated (archived) to another device.

Action: Contact ASG Customer Support.

**DSS4530 VOL=REF=\* NOT RESOLVED**

Explanation: Related IBM message - IEF645I

Action: The referback parameter (DSNAME, DDNAME and/or STEPNAM) value was not resolved. Verify the spelling of names in the referback.

**DSS4531 VOL=REF=\* REFERENCES A DD NOT PREVIOUSLY RESOLVED**

Explanation: Related IBM message - IEF372I

Action: The VOL=REF parameter value was not resolved because it refers to a DD that was not resolved because of RESTART processing.

**DSS4535 VOL=REF=\* CONTAINS INVALID PERIOD**

Explanation: Related IBM message - IEF624I

Action: The referback parameter (DSNAME, DDNAME and/or STEPNAM) contains an invalid period. Check periods and verify symbolic values.

**DSS4540 INVALID DATA SET NAME FORMAT FOR MODEL DSCB**

Explanation: The first positional parameter for DCB, which is currently interpreted as model DSCB, has failed the DSNAME edit.

Action: Contact ASG Customer Support.

**DSS4550 MODEL DSCB DELETED EARLIER IN SYSTEM**

Explanation: The model DSCB existed when the job started, but was deleted before this point.

Action: Contact ASG Customer Support.

**DSS4560 MODEL DSCB NOT FOUND**

Explanation: The model DSCB coded in the DD statement was not found.

Action: It was not built, or the name was misspelled, or the build was unsuccessful.

**DSS4562 Model DSCB must be enclosed in apostrophes**

Explanation: Related IBM messages - IEFC625I IEFC651I

Action: A dataset name that contains a hyphen or parentheses must be enclosed in apostrophes when it is used as a DCB subparameter.

#### DSS4565 DATA SET REF IN DCB HAS BEEN MIGRATED/ARCHIVED

Explanation: The dataset name coded in the DCB= parameter was found in the catalog but the volume serial indicates that it has been migrated (archived) to another device.

Action: Contact ASG Customer Support.

#### DSS4570 DCB=\* NOT RESOLVED

Explanation: The DD statement identified in the DCB parameter referback was not found.

Action: The ddname in the DCB parameter could be misspelled or the DD statement does not exist in the job stream.

#### DSS4572 DCB NOT CODED ON REFERENCED DD STATEMENT

Explanation: The DCB parameter referenced a DD statement that did not contain a DCB parameter.

Action: Contact ASG Customer Support.

#### DSS4575 JCL RECFM INCONSISTENT WITH RECFM ON VOLUME

Explanation: The RECFM parameter coded in the DD statement specifies a value that does not match the value in the volume table of contents (VTOC) for the dataset.

Action: If the program does not override the record format, an open error may occur.

#### DSS4576 JCL LRECL INCONSISTENT WITH LRECL ON VOLUME

Explanation: The LRECL parameter coded in the DD statement specifies a value that does not match the value in the volume table of contents (VTOC) for the dataset.

Action: If the program does not override the logical record length, unpredictable results may occur.

#### DSS4577 JCL BLKSIZE IS SMALLER THAN BLKSIZE ON VOLUME

Explanation: The BLKSIZE parameter coded in the DD statement specifies a value that is less than the BLKSIZE value in the volume table of contents (VTOC) for the dataset.

Action: If the program does not override the block size, an S001 abend may occur (wrong length record).

#### DSS4580 REFERBACK IN DSN OPERAND NOT RESOLVED

Explanation: The DD statement identified in the DSN parameter referback was not found.

Action: Verify the correctness of the DDname and check the spelling.

#### DSS4582 REFERBACK IN DSN OPERAND REFERS TO AN UNNAMED DSNAME

Explanation: Related IBM message - IDG3009I RC=144 Reason Code=14

Action: The DD statement identified in the DSN parameter referback refers to an unnamed dataset name. This could result in a catalog error.

**DSS4585 PGM REFERBACK NOT RESOLVED**

Explanation: The stepname, procstepname, or ddname in the PGM parameter referback was not found.  
Action: Verify the correctness and check the spelling.

**DSS4586 REFERENCED DD STATEMENT DOES NOT DEFINE A PDS MEMBER**

Explanation: The PGM backward reference does not define, as a member of a PDS, a program to be executed.  
Action: Contact ASG Customer Support.

**DSS4590 UNIT AFF NOT RESOLVED**

Explanation: The DD statement identified in the UNIT=AFF parameter was not found within the step.  
Action: Check the name for correctness and spelling.

**DSS4600 PRIMARY SPACE AMOUNT IS MISSING**

Explanation: Related IBM message - IEF646I or IEFC646I  
Action: The SPACE parameter for the new dataset being created did not include a primary amount value. Make sure it is coded and not preceded by a comma.

**DSS4605 Invalid directory request or missing space subparm**

Explanation: Related IBM message - IEF646I or IEFC646I  
Action: The SPACE parameter for the new dataset being created includes primary and/or secondary space for a directory request, or the first SPACE subparameter has been omitted.

**DSS4610 PDS DIRECTORY SPACE AMOUNT IS MISSING**

Explanation: Related IBM message - ABEND SO13-14  
Action: A new partitioned dataset is being created and the SPACE parameter did not contain the PDS directory block allocation amount. Make sure this parameter is coded.

**DSS4615 "" SPACE amount exceeds maximum allowed**

Explanation: Related IBM message - IEF638I or IGD17051I  
Action: The value specified in the SPACE subparameter cannot exceed 16777215. Allocation of a dataset cannot exceed 65535 tracks.

**DSS4650 ACCODE POS 1 IS NOT ALPHA A TO Z, OR IS NOT CAPS.**

Explanation: The first character of the ACCODE field must be alphabetic and capitalized.  
Action: Contact ASG Customer Support.

**DSS4660 ACCODE VALUE IS LONGER THAN 8 CHARACTERS OR OMITTED**

Explanation: The ACCODE value must be 1 to 8 characters in length, starting with an alphabetic character from A to Z.

Action: Contact ASG Customer Support.

#### DSS4670 THE "" PARAMETER HAS AN INVALID VALUE.

Explanation: The value of this parameter is not valid.

Action: See the IBM JCL Reference Manual for valid choices.

#### DSS4675 THE "" NAME IS LONGER THAN 8 CHARACTERS.

Explanation: The name specified for this parameter is longer than eight characters.

Action: It may only be eight characters in length.

#### DSS4676 IMPROPER SUBPARAMETER LIST IN THE "REF" FIELD

Explanation: One of the subparameters of the Volume REF= is not valid for the format being used or a field is out of position with respect to the other subfields in the DD volume parameter.

Action: The REF = has positional parameters and these must be either in the correct sequence, or a comma used to denote their absence.

#### DSS4680 "REFDD" VALUE CAN NOT BE RESOLVED

Explanation: The prior DD reference could not be found in this job's JCL.

Action: Contact ASG Customer Support.

#### DSS4700 TAPE DEVICE REQUIRED FOR OPTCD=Q OR AL OR AUL LABEL

Explanation: This message is issued when the DCB parameter OPTCD=Q or the LABEL parameter AL or AUL has been detected and the device coded on the DD statement is not a tape device. OPTCD=Q and the AL and AUL LABEL parameters indicate an ASCII coded dataset on a tape device.

Action: Contact ASG Customer Support.

#### DSS4710 DCB CODED FOR VSAM DATA SET

Explanation: The DCB keyword was coded in a DD statement that references a VSAM dataset.

Action: AMP is used for VSAM files to complete the information in an Access Method Control Block.

#### DSS4712 NO KEYLEN SPECIFIED FOR VSAM KEY SEQUENCED DATA SET

Explanation: Related IBM message - IGD3009I return code 136 reason code 20

Action: A VSAM sequenced dataset is being defined in JCL but no key length was specified or implied in the DD statement. Specify a KEYLEN parameter in the DD statement or update the DATACLASS SMS construct.

#### DSS4714 KEY SPECIFICATION EXTENDS BEYOND END OF RECORD

Explanation: Related IBM message - IGD3009I return code 304 reason code 0

Action: The key field specified or implied for the VSAM key sequenced dataset defined in this DD statement extends beyond the end of the maximum logical record. Reduce the key length, change the key position, or increase the logical record length.

**DSS4716 AMP CODED FOR NON-VSAM DATA SET**

Explanation: Related IBM message - IEC142I ABEND 116-38

Action: The AMP parameter was coded on a DD statement that references a non-VSAM dataset. Ensure that the DSNAMES specified on the DD statement is correct.

**DSS4720 MAXIMUM NUMBER OF DDNAME REFERENCES EXCEEDED**

Explanation: Related IBM message - IEF631I, IEF654I

Action: A job step or procedure step can contain up to five (5) unresolved DD statements with DDNAME parameters. Each DDNAME parameter must refer to a different DD statement.

**DSS4722 DDNAME REFERENCE REFERS ONLY TO FIRST CONCATENATION**

Explanation: Related IBM message - IEF694I

Action: When the DDNAME reference refers to a concatenation of multiple DDs, only the first DD in the concatenation is actually referenced. The remaining (unlabeled) DDs in the referenced concatenation are concatenated to the labeled DD immediately preceding the concatenation.

**DSS4730 LRECL > BLKSIZE FOR NEW DATA SET**

Explanation: Related IBM message - IEC030, ABEND S002-14, IEC141, ABEND S013-34 or S03B

Action: The DCB specification for the new dataset being created contains conflicting LRECL and BLKSIZE sub-parameters. The LRECL cannot be larger than BLKSIZE.

**DSS4731 STMT NOT RECOGNIZED - SYSIN DD GENERATED BY JOB/SCAN**

Explanation: JOB/SCAN encountered a non-JCL statement in the job stream and, like MVS, has generated a //SYSIN DD statement.

Action: Contact ASG Customer Support.

**DSS4732 CONTROL CARD MEMBER IS ISPF PACKED FORMAT**

Explanation: The Control Card member contains data that is possibly in the ISPF EDIT 'PACK ON' format.

Action: Edit the member and enter the 'PACK OFF' ISPF EDIT profile command. Save the member and resubmit the JOB.

**DSS4733 CONTROL CARD MEMBER IS EMPTY**

Explanation: The member of the Control Card dataset contains no data.

Action: Contact ASG Customer Support.

**DSS4734 CONTROL CARD MEMBER NOT FIXED LENGTH RECORDS**

Explanation: The record format of the Control Card dataset is not fixed

Action: Contact ASG Customer Support.

**DSS4735 CONTROL CARD LIBRARY COULD NOT BE OPENED.**

- Explanation: The control card library, (DSN noted on next line), could not be opened using standard PDS access methods.
- Action: If SUBSYS=LAM or SUBSYS=PANV was coded, the requested interface method was used and an error was found. The Reason Code can be found in message DSS0601 printed in the System Summary Report. A second attempt was then made to open the dataset using standard PDS methods. This message will also appear as a result of an AP Control Table entry specifying a ddname. The text 'CONTROL CARD' will be replaced with the ddname from the AP Control Table entry. The DSS4736 message will also appear.

#### DSS4736 AS SPECIFIED IN THE AP CONTROL TABLE ENTRY

- Explanation: This message will appear following another message which was the result of a parm member search failing.
- Action: The previous message will describe the error that has occurred.

#### DSS4737 AN I/O ERROR OCCURRED ON THE CONTROL STMT DATA SET

- Explanation: An I/O error occurred processing the control statement dataset.
- Action: See message DSS0611 in the JLSSR report which will explain the error.

#### DSS4738 MEMBER xxxxxxxx CREATED IN EARLIER STEP

- Explanation: The control statement member specified could not be listed because it was created in an earlier job step.
- Action: Contact ASG Customer Support.

#### DSS4740 MEMBER "" NOT FOUND IN "" LIBRARY

- Explanation: Related IBM message - IEC141, ABEND S013-18
- Action: The member name specified was not found in the library specified in the JCL ("ABOVE") or in the library specified in the JLJPF DD statement ("JLJPF"). Check the spelling of the library and member name and make sure the member exists.

The JLJPF FILE ALLOCATION statement indicates a dataset name that overrides the control card dataset named in the JCL. An entry must exist in either Control Table AB referencing the ddname to be overridden or Control Table AQ referencing the dsname to be overridden. Also, the ACCESS CONTROL CARDS Run Options must be set to "Yes".

This message will also appear as a result of a DDNAME being specified on an AP Control Table entry. The member name as specified in the PARM string could not be found in the library specified by the ddname on the AP Control Table entry. The DSS4736 message will also appear.

#### DSS4741 MEMBER SPECIFICATION INVALID FOR DSORG=PS

- Explanation: Related IBM message - IEC141, ABEND S013-14
- Action: A member was specified for a sequential dataset (DSORG=PS ). A member can only be specified if the dataset is partitioned (DSORG=PO or POU).

#### DSS4742 CONTROL CARD MEMBER membername FOUND IN DATA SET:

- Explanation: The control card member 'membername' was read from the file or files allocated with the DDNAME of JLJPF and not necessarily the dataset specified in the JCL stream.
- Action: This message is immediately followed by a DSS8900 message naming the dataset from which the control cards were actually read.

#### DSS4750 "membername" NOT VERIFIED - LIBRARY NOT ALLOCATED

- Explanation: The library in this DD could not be opened.
- Action: The dataset is probably in use by another user with DISP=OLD. This message will also appear as a result of a DDNAME being specified on an AP Control Table entry. The text 'LIBRARY' will be replaced with a ddname which could not be allocated. Check your FILE ALLOCATION statements to ensure the ddname matches the AP table entry. This message will be followed by a DSS4736 message.

#### DSS4755 ACCESS DENIED BY SECURITY

- Explanation: JOB/SCAN has attempted to open the dataset for input (access control cards, program verification, etc.) and access was denied by the security authorization system (RACF, ACF2, etc.).
- Action: Contact ASG Customer Support.

#### DSS4757 "" ACCESS TO DATA SET DENIED TO USER ""

- Explanation: The security exit (RACF, ACF2, etc.) has checked the dataset for access privilege and the particular access (READ, UPDATE, or ALTER) has been denied by the system.
- Action: Contact ASG Customer Support.

#### DSS4759 DSNAME SECURITY VALIDATION SUSPENDED FOR THIS JOB

- Explanation: The "DSNAME" parameter has been coded on the "SECURITY EXIT=" Run Option and the security exit (RACF, ACF2, etc.) has passed back a non-zero return code when validating the USERID and/or GROUP name from the JOB statement (or // \*LOGONID statement) of the job currently being scanned.
- Action: Dsname security validation will be attempted again when JOB/SCAN begins scanning the next JOB statement in the input stream. If you are using a ASG supplied exit, a message is displayed following this message that explains the reason for the non-zero return code. See [Security Interfaces](#) for more information.

#### DSS4800 DSORG "" INVALID WITH DEVICE ""

- Explanation: The DSORG value is not valid for the device type specified.
- Action: Contact ASG Customer Support.

#### DSS4802 CREATION OF ISAM DATA SETS IS NO LONGER SUPPORTED

- Explanation: As of z/OS 1.7, the system no longer supports creation or opening of indexed sequential datasets.
- Action: Contact ASG Customer Support.



**DSS4860 PDS CONCATENATED WITH TAPE**

Explanation: A PDS cannot be concatenated with a tape dataset.

Action: Contact ASG Customer Support.

**DSS4880 LRECL "" CANNOT BE CONCAT. WITH LRECL ""**

Explanation: Related IBM message - IEC020I, IEC026I, ABEND S001-1 or S637

Action: The record lengths in the error message cannot be concatenated. Concatenated datasets must be compatible.

**DSS4900 "" IS INVALID IN SPACE DESCRIPTION**

Explanation: Related IBM message - IEF644I

Action: The string in quotes is not a valid space description for the SPACE parameter. Check the spelling and values coded.

**DSS4905 BLOCK LENGTH IN SPACE PARM EXCEEDS 65535 BYTES**

Explanation: Related IBM messages - IEF638I, IGD17006I

Action: The average block length specified for the space allocation of this dataset exceeds the allowable maximum, which is 65535 bytes. Correct the average block length that you specified in either the JCL or the data class.

**DSS4910 "" IS MUTUALLY EXCLUSIVE WITH ""**

Explanation: The two keywords in quotes are mutually exclusive.

Action: Contact ASG Customer Support.

**DSS4930 UNIT COUNT LIMIT EXCEEDED - ALLOWED MAX IS 59**

Explanation: Related IBM message - IEF638I

Action: The UNIT COUNT field may not contain a number larger than 59.

**DSS4940 UNABLE TO IDENTIFY UNIT NAME ""**

Explanation: Related IBM message - IEF210I

Action: The value coded for the UNIT parameter cannot be identified as a valid device type for this installation.

**DSS4950 GDG NAME LONGER THAN 35 CHARACTERS**

Explanation: The maximum length for a generation dataset group is 35 characters, including the periods.

Action: Contact ASG Customer Support.

**DSS4960 A TAPE DEVICE SHOULD NOT BE CODED WITH "AMP"**

Explanation: Because "AMP" is a parameter for VSAM files, any tape device coded would be incorrect.

Action: Contact ASG Customer Support.

**DSS4965 VOLUME "volid" IS A SCRATCH VOLUME**

Explanation: The volume specified in the VOL=SER= parameter is within a common scratch pool in the library and cannot be explicitly mounted using the volume serial number.

Action: Contact ASG Customer Support.

**DSS4970 DATA SET ALREADY EXISTS ON VOLUME**

Explanation: Related IBM message - IEF253I

Action: A new dataset being created on the target volume already exists. Check the DISP and the spelling of the dataset name.

**DSS4975 VOLUME " " NOT AVAILABLE**

Explanation: While reading the VTOC, a return code was received indicating the volume was not mounted.

Action: Contact ASG Customer Support.

**DSS4980 VALUE OF BUFL SUBPARAMETER IS NOT NUMERIC**

Explanation: Must be a numeric value with a maximum value of 32,760.

Action: Contact ASG Customer Support.

**DSS4990 BLKSIZE INCOMPATIBLE WITH LRECL**

Explanation: The block size coded is not compatible with the associated record length.

Action: Contact ASG Customer Support.

**DSS4995 INVALID BLKSIZE SPECIFIED WITH RECFM=" "**

Explanation: The block size coded is invalid with the RECFM specified.

Action: Contact ASG Customer Support.

**DSS5010 INVALID DATA SET NAME FORMAT FOR VOL=REF**

Explanation: The parameter VOL=REF=dsname was coded, but the DSNAME failed the edit test.

Action: When the dsname includes special characters other than periods, enclose it in apostrophes. A dsname cannot be a GDG name or a GDG member.

**DSS5020 FORMS PARAMETER IS TOO LONG**

Explanation: The special FORMS number coded in the SYSOUT= sub-parameter contains more than four characters.

Action: Contact ASG Customer Support.

**DSS5022 CHARS PARAMETER IS TOO LONG**

Explanation: The CHARS parameter specified contains more than four characters.

Action: Contact ASG Customer Support.

**DSS5027 INVALID OVERLAY NAME IN FLASH PARAMETER**

- Explanation: The overlay name specified in the FLASH parameter must be one through four alphameric or national (\$,#,@) characters.
- Action: A null subparameter is invalid in a FLASH parameter on a DD statement, but is permitted on an OUTPUT JCL statement.

**DSS5028 INVALID COUNT IN FLASH PARAMETER**

- Explanation: The count subparameter of the FLASH parameter must be a number between 0 and 255.
- Action: Contact ASG Customer Support.

**DSS5030 UNABLE TO IDENTIFY UNIT**

- Explanation: The UNIT type associated with this DD statement could not be determined by JOB/SCAN because a VOLSER was coded and bypassed the system catalog.
- Action: Contact ASG Customer Support.

**DSS5032 VOLUME "" INCONSISTENT WITH UNIT ""**

- Explanation: Related IBM messages - IEF210I, IEF245I, IEF702I
- Action: The UNIT name specified by the UNIT parameter is not valid for the volume serial number specified in the previous DD statement. The device on which the volume is currently mounted is inconsistent with the device specified.

ADDITIONAL NOTE: This error may also indicate that the VOLUME catalog device information does not agree with the tape management system's (TMS or CA-1) TMC volume device code or that SHORT DESCRIPTION RUN OPTION has defaulted (or been coded) to JCL and that an override (+I,O,R,etc) statement is on the DD card.

**DSS5034 VOLUME "" CANNOT BE MOUNTED ON UNIT ""**

- Explanation: Related IBM message - IEF482I
- Action: The volume specified cannot be mounted because the unit address specified in the UNIT parameter contains a permanently resident or reserved volume.

**DSS5040 NO MODEL DSCB CODED**

- Explanation: The model DSCB must be coded on all new generations of a GDG.
- Action: This message can also be caused by coding DCB=\*.xxx (backwards reference is not allowed for GDG datasets).

**DSS5080 GENERATION DOES NOT EXIST ON VOLUME**

- Explanation: The catalog for this generation of a GDG refers to a DASD unit that does not contain the dataset.
- Action: Contact ASG Customer Support.

**DSS5085 GDG INVALID AS REFERENCED IN VOL=REF**

- Explanation: A generation data group name or a relative generation number cannot be used with VOL=REF=.
- Action: Contact ASG Customer Support.

**DSS5110 FILE NOT FOUND ON DASD VOLUME**

Explanation: The JCL or catalog referred to a volume that did not contain the data set.  
Action: Contact ASG Customer Support.

**DSS5120 DATA SET CANNOT BE FOUND**

Explanation: Related IBM message - IEF212I  
Action: The dataset cannot be found because it has not been passed, nor is it in the catalog.

**DSS5130 FILE IS ALREADY CATALOGED**

Explanation: Related IBM messages - IEF287I, IGD17101I  
Action: An attempt was made to catalog an already cataloged dataset. This may result in the IBM message "NOT CATALOGED 2".

**DSS5131 WARNING - FILE IS ALREADY CATALOGED**

Explanation: Possible causes for this message are: - the file was created in an earlier step - the file was created by an earlier job but never deleted  
Action: Contact ASG Customer Support.

**DSS5132 FILE NOT RECATALOGED**

Explanation: Related IBM messages - IEF287I, IEF377I  
Action: The dataset to be cataloged had been cataloged previously. Either a catalog entry already exists for the DSNNAME specified, or a catalog entry for an ALIAS of another dataset matches the DSNNAME specified.

This condition usually results in the IBM informational message IEF287I "NOT CATALOGED 2" being issued. If your installation has coded the "CATLG-ERR FAILJOB(YES)" in the ALLOCxx member of SYS1.PARMLIB, you may want to change the severity of this message in the JOB/SCAN Control Table AW. If not, you may want to suppress this message.

**DSS5133 FILE NOT UNCATALOGED**

Explanation: Related IBM message - IEF287I  
Action: The dataset to be uncataloged is either an SMS-Managed dataset, a VSAM dataset or is a non-SMS-Managed dataset with VOLUME=SER or VOLUME=REF coded on the DD statement. In these three cases, the UNCATLG disposition is treated as KEEP.

**DSS5135 DATA SET GENERATION NOT CATALOGED - GDG NOT FOUND**

Explanation: Related IBM message - IEF287I  
Action: The dataset name has the same low level qualifier (GnnnnVnn) as a GDG (generation data group) generation but the GDG was not found in the catalog.

**DSS5136 CATALOG CONTAINS NO GENERATIONS FOR THIS GDG**

Explanation: The generation data group contains no generation datasets.

Action: This message is issued when the JCL specifies a GDG base name with a DISP parameter of (OLD or SHR,DELETE) and there are no generations found in the catalog to delete.

#### DSS5140 WARNING - DISPOSITION MAY CAUSE UNEXPECTED RESULTS

Explanation: This message refers to instances when a JOB will execute but may produce undesirable results such as creating two versions of the same dataset on two volumes or a dataset sequence that leaves the catalog pointing to the wrong volume.

Action: Contact ASG Customer Support.

#### DSS5145 MAXIMUM DD STATEMENTS ALLOWED PER STEP EXCEEDED

Explanation: MVS allows a maximum of 3273 DD statements per program step.

Action: Contact ASG Customer Support.

#### DSS5150 CONCATENATION LIMIT EXCEEDED

Explanation: There is a limit to how many DD statements are allowed in a partitioned concatenation.

Action: Add together the number of PDS extents, the number of PDSE's and the number of UNIX directories in the concatenation. The sum cannot exceed 255. For example, you can concatenate 15 PDS's of 16 extents each with 8 PDSE's and 7 UNIX directories ((15 x 16) + 8 + 7 = 255 extents).

#### DSS5160 CANNOT CONCATENATE DUMMY WITH DSORG -

Explanation: DUMMY can only be a member of a concatenation for sequential access.

Action: Contact ASG Customer Support.

#### DSS5170 WARNING - ALL FILES AFTER A DUMMY WILL BE IGNORED

Explanation: A DUMMY may be added to a group of concatenated datasets, but all of those that follow the DUMMY will not be referenced.

Action: Contact ASG Customer Support.

#### DSS5175 DUMMY MUST BE CODED BEFORE OTHER DD PARAMETERS

Explanation: DUMMY must be coded first if coded with other DD parameters.

Action: Contact ASG Customer Support.

#### DSS5180 WARNING - POSSIBLE ERROR IN CONCATENATION

Explanation: This error indicates an apparent concatenation of dataset names and flags it as possibly incorrect.

Action: Contact ASG Customer Support.

#### DSS5190 DSORG " " CANNOT BE CONCATENATED WITH " "

Explanation: You cannot mix and match DSORG types in a concatenation.

Action: The most common erroneous combinations are Partitioned (PO) and Sequential (PS).

#### DSS5200 RECFM " " CANNOT BE CONCATENATED WITH " "

- Explanation: Two or more datasets are concatenated and they have record formats that are not compatible.
- Action: Contact ASG Customer Support.

**DSS5201 RECFM "FB" CANNOT BE CONCATENATED WITH RECFM "F"**

- Explanation: A dataset with a RECFM of 'FB' is concatenated to a dataset with a RECFM of 'F'.
- Action: This concatenation is not allowed by the program being executed.

**DSS5209 UNABLE TO IDENTIFY VOLUME FOR A JOBLIB/STEPLIB**

- Explanation: This warning is generated when a non-specific volume request is made for a JOBLIB or STEPLIB DD statement. A non-specific volume request is a DD statement for a new data that can be assigned to any volume or volumes. JOB/SCAN is not able to determine which volume will be selected at the time the job is run, therefore, JOB/SCAN is unable to determine if the dataset exists or not.
- Action: This message may be generated when a STEPLIB or a JOBLIB DD in a procedure is overridden with a cataloged dataset name and the procedure refers to a non-specific volume request but the volume and unit override parameters were not specified as null.

**DSS5210 IMPROPER ATTRIBUTES FOR A JOBLIB/STEPLIB**

- Explanation: This error is generated if one or more of the following conditions is met.
- Action: The dataset does not reside on a direct access device. The dataset is not on the volume. The dataset is not a partitioned dataset. The dataset record format is not "U".

**DSS5215 CONCATENATED DATA SETS HAVE UNLIKE ATTRIBUTES**

- Explanation: Related IBM error message is IEC0261 (ABEND S637-0C).
- Action: This error is generated if concatenated datasets have unlike device attributes. If DCBOFLGS bit 4 is set by the problem program, then this concatenation may be legal.

**DSS5220 INVALID SYSOUT WRITER NAME**

- Explanation: Identifies an external writer.
- Action: The name is one to eight alphanumeric characters and must start with an alphabetic or national character. Do not code STDWTR as a writer name. For JES3 systems, do not code NJERDR.

**DSS5225 INVALID DSNAME FOR IN-STREAM OR SYSOUT DATA SET**

- Explanation: Related IBM message - \$HASP131, \$HASP132, \$HASP133, \$HASP134 IAT4250, IAT4251, IAT4252, IAT4253 IEF655I
- Action: The dataset name for in-stream or sysout datasets consists of two ampersands (&&) followed by one through eight alphanumeric or national (\$,#,@) characters, a hyphen, or a character X'CO'. The first character following the two ampersands must be alphabetic or national (\$,#,@). Do not code the following dataset names; they are reserved for system use. JESJCL JESJCLIN JESMSG LG JESYSMSG

**DSS5229 VALUE OF SMBVSP NOT NUMERIC**

Explanation: The value of the SMBVSP parameter must be between 1 to 2048000 kilobytes or 1 to 2048 megabytes.

Action: Contact ASG Customer Support.

**DSS5230 VALUE IN AMP MUST BE ENCLOSED IN QUOTES**

Explanation: When the parameter contains more than one sub-parameter, separate the sub-parameters with commas and enclose the sub-parameter list in apostrophes inside the parentheses.

Action: Contact ASG Customer Support.

**DSS5231 SMBVSP PARAMETER MUST END IN "K" OR "M"**

Explanation: The SMBVSP parameter must end in "K" for kilobytes or "M" for megabytes.

Action: Contact ASG Customer Support.

**DSS5232 SMBVSP SIZE EXCEEDS 2048M**

Explanation: The SMBVSP parameter specifies a size between 1 and 2048 megabytes.

Action: Contact ASG Customer Support.

**DSS5233 VALUE IN ACCODE MUST NOT BE ENCLOSED IN QUOTES**

Explanation: Quotes are not needed and may not be used in the ACCODE parameter.

Action: Contact ASG Customer Support.

**DSS5234 SMBVSP SIZE EXCEEDS 2048000K**

Explanation: The SMBVSP parameter specifies a size between 1 and 204800 kilobytes.

Action: Contact ASG Customer Support.

**DSS5235 BLANK VALUE FOR SYSOUT IS NOT ALLOWED**

Explanation: No CLASS value was found in the SYSOUT statement or on an associated OUTPUT statement.

Action: Contact ASG Customer Support.

**DSS5240 BLOCKSIZE IS GREATER THAN 32760**

Explanation: The maximum allowed block size is 32760.

Action: Contact ASG Customer Support.

**DSS5250 VALUE OF SYSOUT CLASS IS TOO LONG**

Explanation: The SYSOUT CLASS can only be one character long.

Action: Contact ASG Customer Support.

**DSS5260 EXTRANEIOUS PROC STATEMENT FOUND IN PROC**

Explanation: A misplaced PROC statement causes this error.

Action: Contact ASG Customer Support.

**DSS5270 "" IS INVALID VALUE FOR "".**

Explanation: The value of the specified keyword has exceeded its permissible limits.

Action: For example, the maximum size of LRECL is 32760. If "LRECL=33000" was coded, this error would be generated.

**DSS5280 INVALID FILE SEQUENCE # - STACKING ON SCRATCH TAPE**

Explanation: Your LABEL parameter specifies a dataset sequence number that is greater than 1 and you have not supplied a specific volume serial number.

Action: Contact ASG Customer Support.

**DSS5284 MVS CATALOG/TMS DATABASE SHOWS DATA SET ON FILE nnnn**

Explanation: Related IBM message - IEC149I, ABEND S813-04

Action: Your LABEL parameter specifies a dataset sequence number that is not the same as the sequence number found in either the MVS system catalog or the CA1, RMM, or ZARA database.

**DSS5285 MULTIFILE SEQUENCE ERROR**

Explanation: You are trying to create a dataset using an incorrect dataset sequence number in your LABEL parameter. Dataset sequence numbers must be specified in consecutive order in increments of 1.

Action: For example, if you already have created datasets 1 and 2, you must use a dataset sequence number of 3 in your LABEL parameter when creating the next dataset in the volume. You may not skip to dataset sequence number 4.

**DSS5286 EXISTING FILE WILL BE OVERWRITTEN ON MULTIFILE TAPE.**

Explanation: The file sequence number is associated with an existing dataset.

Action: The existing dataset will be overwritten.

**DSS5287 MIXED EXPIRATION DATES ON MULTIFILE TAPE.**

Explanation: Expiration date of a secondary dataset has been written to a volume with date different from that used for the first file.

Action: If one runs the job on MVS with the Computer Associates-Tape Monitor System (CA-TMS), the corresponding message issued by CA-TMS is: IECTMS10, WARNING, A MIXED EXPIRATION WAS ISSUED FOR jobname,stepname,ddname,dddd,vvvvv,dsname Interface with the CA-TMS application is specified by using the Problem Program Interface (PPI) option of JOB/SCAN as described in [Program Product Interface \(PPI\)](#). Issuance of DSS5287 is independent of the CA-TMS option used (CA1, CA1R5 or none specified). \*

**DSS5290 DUAL DEN NOT CATALOGED IF NOT OPENED**

Explanation: An attempt was made to catalog a dual density tape file using IEFBR14.

Action: Contact ASG Customer Support.



**DSS5300 INVALID VALUE FOR RECFM**

Explanation: The most common values for RECFM are:

U - Undefined-length records

V - Variable-length records

F - Fixed-length records

B - Blocked records

T - Track overflow used

A - ASA control char is the first byte of data

M - Machine control char is the first byte of data

For a complete list of RECFM values and the permissible combinations refer to the IBM JCL Reference Manual.

**DSS5310 DSN = data.set.name**

Explanation: The dataset name associated with the following DSS53xx message.

Action: Contact ASG Customer Support.

**DSS5311 NOT FOUND ON VOLUME xxxxxx**

Explanation: The RMM database was accessed for a tape dataset and the dataset was not found on the volume specified in the DD statement or indirectly by the Catalog entry.

Action: It is possible that the DD card is incorrect or that the tape dataset has been deleted or replaced.

**DSS5312 NOT FOUND ON VOLUMES**

Explanation: The RMM database was accessed for a multivolume tape dataset and the dataset was not found on the volumes specified in the DD statement or indirectly by the Catalog.

Action: It is possible that the DD statement is incorrect or that the tape dataset has been deleted or replaced.

**DSS5315 ACTUALLY FOUND ON VOLUME xxxxxx**

Explanation: A search of the RMM database was done in an attempt to locate the dataset.

Action: The dataset was found on the named volume.

**DSS5317 VOLUME xxxxxx IS NOT AN RMM MANAGED VOLUME**

Explanation: The volume specified directly in a DD statement or indirectly in the Catalog was not found in the RMM database.

Action: Contact ASG Customer Support.

**DSS5318 VOL SEQ ERROR. JCL EXPECTED xxxxxx, RMM FOUND yyyyyy**

Explanation: A multivolume dataset is requested but the order of the volumes appears to be out of sequence.

Action: The first volume serial was specified by either the JCL or the Catalog. The second volume serial was the next volume in the sequence according to the RMM database.

**DSS5319 RMM SHOWS TAPE xxxxxx AT STOR LOC=yyyyyyyy**

- Explanation: RMM shows that the volume referenced by the DD statement is not currently in the library.
- Action: It is either at a remote storage location or at a loan location and therefore unavailable for a mount. Check with your librarian concerning the location of the volume.

**DSS5320 RMM SHOWS VOLUME xxxxxx IN yyyyyyy STATUS**

- Explanation: The RMM database was accessed for a tape volume and the volume is not available to be mounted. RMM will not allow a tape in SCRATCH, INIT or ENTRY status to be mounted for input.
- Action: If a volume has been SCRATCHED, it is possible that its expiration date had been reached and it had been replaced or overwritten.

**DSS5325 RMM SEARCHDS FAILED RC=xxxx, RSN=yyyy**

- Explanation: An unexpected return code has been received after an RMM database access.
- Action: Contact ASG Customer Support.

**DSS5326 RMM LISTVOLUME FAILED RET=xxxx, RSN=yyyy**

- Explanation: An unexpected return code has been received after an RMM database access.
- Action: Contact ASG Customer Support.

**DSS5509 NON-ZERO RC FROM STANDARDS EXIT - NO ASSOCIATED MSG**

- Explanation: Standards program specified a return code other than zero, but there was no message specified to print on the Structured JCL Listing.
- Action: This message is printed by JOB/SCAN when no message has been returned by the Standards program and the return code received from the Standards program is not zero.

**DSS5510 PROCESSING TERMINATED - STANDARDS EXIT RETURN CODE**

- Explanation: Standards program specified a return code greater than 19.
- Action: This tells JOB/SCAN to discontinue processing. Consult the JOB/SCAN Structured JCL Listing to see specific messages at the line(s) of code that is in error.

**DSS5514 INVALID ACTION "" FROM "" FOR ""**

- Explanation: The named User Exit specified an action for this message number that was not 'N' (None), 'C' (Change severity), or 'S' (Suppress message).
- Action: Verify that your User Exit is passing back a correct action flag to JOB/SCAN for all error messages. Refer to the EPL-JOB-SCAN-ERRORS section of the Standards Exit.

**DSS5520 MEMBER "" NOT FOUND IN LIBRARY**

- Explanation: User Text was requested for this report but the specified member name was not found in the User Text Library.

Action: This error most often occurs when you do not have User Text members but the option is enabled.

TEXTENTITY parameters of various REPORT OPTIONS enable User Text. Valid sub-parameters of TEXTENTITY are JOB,PROC,PGM,FILE. Search the REPORT OPTION section in PARMLIB for any unwanted TEXTENTITY= parameters or sub-parameters and disable them.

For example:

To eliminate the search for FILE level members but to allow the search for PROC level members when including User Text in the SJL Report, change:

REPORT OPTIONS

SJL=TEXTSOURCE=FILE,TEXTENTITY=(JOB,PROC,PGM,FILE)

to:

REPORT OPTIONS

SJL =TEXTSOURCE=FILE,TEXTENTITY=(JOB,PROC,PGM)

Reports that may include the TEXTENETITY parameter are:

3. Structured JCL Listing 9. Flowcharts
4. Data Set XREF 10. Job Documentation
6. Program XREF 11. Tape Usage Analysis
7. PROC XREF 14. Report Analysis
8. I/O Listing

#### DSS5530 INCORRECT COMMAND - STATEMENT IGNORED

Explanation: IDCAMS. The first word in the command statement is not a valid IDCAMS command.

Action: Contact ASG Customer Support.

#### DSS5531 INCORRECT COMMAND " " - STATEMENT IGNORED

Explanation: IDCAMS. The command found is not a valid IDCAMS command.

Action: Contact ASG Customer Support.

#### DSS5535 ABOVE TEXT BYPASSED TILL NEXT COMMAND FOUND

Explanation: IDCAMS. A valid IDCAMS command was not found and caused text to be by-passed until a valid command was found.

Action: Contact ASG Customer Support.

#### DSS5536 TEXT/DATA BYPASSED TILL NEXT STATEMENT FOUND IDCAMS.

Explanation: An error was found in an IDCAMS statement causing the remainder of the text or data of that statement to be by-passed until the next valid statement was found.

Action: Contact ASG Customer Support.

#### DSS5540 REQUIRED xxxx PARAMETER NOT FOUND

Explanation: The specified parameter has been omitted and is required for the operation requested.

Action: Contact ASG Customer Support.

#### DSS5550 EXCESSIVE RIGHT PARENTHESES FOUND

Explanation: IDCAMS. Statement contained more right parentheses than left parentheses.

Action: Contact ASG Customer Support.

#### DSS5560 NO VERB FOUND - STATEMENT IGNORED

Explanation: The scan was looking for a verb but instead received a zero length entry or a blank line.

Action: This might occur if the first character encountered was a comma.

#### DSS5565 PRECEDING COMMAND BYPASSED DUE TO LASTCC/MAXCC COND

Explanation: Related IBM message - IDC0204I

Action: The preceding command was checked for syntax errors but was not executed because "IF" processing determined that conditions were not met.

#### DSS5566 All remaining commands flushed - LASTCC/MAXCC > 15

Explanation: Related IBM message - IDC3207I

Action: All remaining commands in this input stream are bypassed because the value of the LASTCC or MAXCC has exceeded 15. Syntax checking continues but no "processing" is done.

#### DSS5568 All remaining commands cancelled

Explanation: Related IBM message - IDC0254I

Action: All remaining commands in this input stream are bypassed because the CANCEL command was executed. Syntax checking continues but no "processing" is done.

#### DSS5570 ILLEGAL CHARACTER FOUND IN STATEMENT SCAN

Explanation: IDCAMS. Internal processing error.

Action: Contact ASG Customer Support.

#### DSS5572 DCB MISSING FROM REPRO OUTFILE DD STATEMENT

Explanation: IDCAMS. The OUTFILE parameter on the REPRO command refers to a DD statement that contains no DCB information.

Action: This may cause IDCAMS to be unable to open the dataset for output.

#### DSS5580 EXPECTED CONTINUATION NOT RECEIVED

Explanation: IDCAMS. Previous statement ended with a continuation character but an end of file or a new statement was encountered instead of a continuation.

Action: A hyphen as the last non-blank character before the right margin indicates continuation of the command. A plus sign as the last non-blank character indicates a continuation of the value within the command. The DO-END sequence does NOT require continuation characters.

#### DSS5581 COMMENT STARTED WITH "/\*" DOES NOT END WITH "\*/"

Explanation: IDCAMS. JOB/SCAN was unable to find the end of a comment statement within an idcams sysin stream before encountering the start of the next IDCAMS command.

Action: Contact ASG Customer Support.

#### DSS5582 MORE THAN ONE VOLUME SPECIFIED

Explanation: IEHPROGM. For SCRATCH or RENAME, if VTOC or MEMBER is specified, VOL cannot specify more than one volume.

Action: Contact ASG Customer Support.

#### DSS5583 VOLUME LIST TRUNCATED TO ONE VOLUME

Explanation: IEHPROGM. Only the VTOC of the first volume specified will be scratched.

Action: If more than one volume is to be scratched, there must be one SCRATCH VTOC statement coded for each volume.

#### DSS5584 Invalid/Unrecognized statement - bypassed

Explanation: IEHPROGM. JOB/SCAN found a command that is not a valid IEHPROGM command.

Action: The statement is bypassed.

#### DSS5585 Invalid continuation - cc1 to cc15 not blank

Explanation: IEHPROGM. Continuation utility control statements must begin in cc16.

Action: Columns 1 to 15 must be blank.

#### DSS5586 Invalid continuation - must start in cc16

Explanation: IEHPROGM. The first character of a utility continuation statement must be in cc16.

Action: Contact ASG Customer Support.

#### DSS5587 For continuation, cc72 must not be blank

Explanation: IEHPROGM. CC72 of a continued utility control statement must not be blank.

Action: Any non-blank character is acceptable.

#### DSS5588 Invalid non-blank character in cc72

Explanation: IEHPROGM. CC72 of a non-continued utility control statement must be blank.

Action: The entire statement must be coded from cc1 to cc71.

#### DSS5589 Member "" not found

Explanation: IDCAMS. The member specified in a DELETE or an ALTER control statement was not found in the PDS specified. IEHPROGM. The member specified in a SCRATCH or a RENAME control statement was not found in the PDS specified.

Action: Contact ASG Customer Support.

#### DSS5590 "" INVALID WITH ""

Explanation: IDCAMS. The keywords within the quotation marks are incompatible.

Action: Contact ASG Customer Support.

#### DSS5591 NEWNAME "" ALREADY EXISTS - NOT RENAMED

Explanation: The member name specified in the NEWNAME parameter currently exists in the dataset.

Action: The member will not be renamed.

#### DSS5592 Value required for "" keyword

Explanation: IDCAMS. The subparameter list or data required for this keyword is missing.

Action: TEB

#### DSS5593 DATA SET IS NOT A PDS

Explanation: IEHPROGM. The dataset specified in a SCRATCH or a RENAME control statement with the MEMBER keyword specified is not a partitioned dataset.

Action: TBE

#### DSS5594 DATA SET IS NOT A PDS

Explanation: IDCAMS. The dataset specified in the ALTER control statement renaming a member or the DELETE control statement deleting a member is not a partitioned dataset.

Action: Contact ASG Customer Support.

#### DSS5595 VALUE IS TOO LONG

Explanation: IDCAMS. The value entered contained more characters than the field allowed. DSS5596 NON-SPECIFIC VOLSER MUST FOLLOW SPECIFIC VOLSER IDCAMS. Invalid volume list. In a volume list, all non-specific volsers appear after all specific volsers.

Action: Contact ASG Customer Support.

#### DSS5598 HEX value is incorrect length

Explanation: IDCAMS. The value entered for a hex string contains an uneven number of characters.

Action: You must have two characters representing each hex byte. (0-9,A-F) ie. X'C1C2' NOT x'C1C'

#### DSS5600 VALUE IS NOT NUMERIC

Explanation: IDCAMS. The value entered is not a numeric value.

Action: Contact ASG Customer Support.

#### DSS5605 VALUE OUT OF RANGE FOR GDG LIMIT

Explanation: IDCAMS. A GDG limit may not exceed 255. Refer to your IBM manual for further assistance.

Action: Contact ASG Customer Support.

#### DSS5610 INVALID DATE FORMAT - MUST BE JULIAN

Explanation: IDCAMS. The date entered was not in Julian format.

Action: Contact ASG Customer Support.

**DSS5620 NO VALUE GIVEN FOR VARIABLE**

Explanation: IDCAMS. Keyword parameter did not have a required sub-parameter.

Action: Contact ASG Customer Support.

**DSS5630 MISSING RIGHT PARENTHESIS AFTER VALUE**

Explanation: IDCAMS. Expected right parenthesis not found where required.

Action: Contact ASG Customer Support.

**DSS5640 MISSING LEFT PARENTHESIS BEFORE VALUE**

Explanation: IDCAMS. Expected left parenthesis not found where required.

Action: Contact ASG Customer Support.

**DSS5641 LEFT PARENTHESIS MISSING AFTER KEYWORD "keyword"**

Explanation: IDCAMS. An opening parenthesis delimiting the subparameter list or constants associated with the keyword does not precede the keyword.

Action: Contact ASG Customer Support.

**DSS5650 "" IS NOT VALID FOR THIS COMMAND**

Explanation: Related IBM message - IDC3211I

Action: The named keyword parameter is not recognized in the context of the IDCAMS command being processed. It may be misspelled, not applicable, or specified as a subparameter in the wrong subparameter list.

**DSS5655 INCORRECT SPECIFICATION OF SPACE ALLOCATION**

Explanation: Related IBM message - IDC3505I IDCAMS.

Action: CYLINDERS, TRACKS and/or RECORDS are specified improperly for this command. Refer to your IDCAMS manual.

**DSS5656 Average record size exceeds Maximum record size**

Explanation: Related IBM message - IDC3515I IDCAMS.

Action: The first parameter in the RECORDSIZE specification (Average record size) is greater than the second parameter in the RECORDSIZE specification (Maximum record size). Refer to your IDCAMS manual.

**DSS5660 COMMAND STRUCTURE INVALID - REMAINDER SKIPPED**

Explanation: IDCAMS. Command list ending values of ")" or ";" were encountered, or no commands were found in the stack.

Action: Contact ASG Customer Support.

**DSS5664 keyword IGNORED - UNSUPPORTED IN z/OS**

Explanation: IDCAMS. The named keyword is no longer supported in z/OS. The keyword is ignored by z/OS. The severity of this message may be downgraded to zero by using the Standards Facility or the 'AW' Control Table.

Action: It is not an error but the keyword should be removed from the JCL.

#### DSS5670 DUPLICATE KEY :

Explanation: IDCAMS. The same keyword parameter was entered more than once.

Action: Contact ASG Customer Support.

#### DSS5680 REQUIRED POSITIONAL PARAMETER NOT FOUND

Explanation: IDCAMS. The IDCAMS command syntax requires a positional parameter.

Action: Contact ASG Customer Support.

#### DSS5690 EXCESSIVE NUMBER OF LEFT PARENTHESES FOUND

Explanation: IDCAMS. Statement contained more left parentheses than right parentheses.

Action: Contact ASG Customer Support.

#### DSS5695 DDNAME " " NOT FOUND IN STEP

Explanation: The DD name referenced in DDNAME= was not resolved.

Action: Contact ASG Customer Support.

#### DSS5697 VOLUME SERIAL NUMBER " " IS INCORRECT

Explanation: The VOL parameter specifies an incorrect volume serial number. The VOL parameter must match the volume serial number specified on the DD statement that the FILE parameter refers to. DSS5700 FOUND " " - EXPECTING " " IDCAMS. While processing a SET command, JOB/SCAN found the value in the first pair of quotation marks, while the command expected the value in the second pair of quotation marks.

Action: Contact ASG Customer Support.

#### DSS5710 FOUND " ", NOT MAXCC OR LASTCC

Explanation: IDCAMS. In an IF statement, JOB/SCAN found the value in the quotation marks while the statement expected MAXCC or LASTCC.

Action: Contact ASG Customer Support.

#### DSS5720 FOUND " ", NOT THEN

Explanation: IDCAMS. The syntax in an IF statement requires "THEN" instead of the value in the quotation marks.

Action: Contact ASG Customer Support.

#### DSS5730 NULL VALUE FOUND IN IF STATEMENT

Explanation: After an IF, either the relationship of the value was returned as null or was non-existent.



Action: Contact ASG Customer Support.

#### DSS5740 IF STATEMENT PREMATURELY ENDED

Explanation: IDCAMS. The required parameters of the IF statement were not present.

Action: Contact ASG Customer Support.

#### DSS5750 INVALID RELATIONAL IN IF STATEMENT

Explanation: IDCAMS. The IF statement contained an invalid relational operator.

Action: Contact ASG Customer Support.

#### DSS5755 AN "ELSE" COMMAND APPEARS IMPROPERLY

Explanation: Related IBM message - IDC4227I

Action: An "ELSE" command appears without a matching IF - THEN command. Check your command continuations (hyphens) and make sure you have ended each DO group with an END command.

#### DSS5760 "" IS ADDITIONAL POSITIONAL PARAMETER

Explanation: IDCAMS. An additional, illegal positional parameter was found in an IDCAMS statement.

Action: Contact ASG Customer Support.

#### DSS5770 "" is an invalid value for SORTMESSAGELEVEL

Explanation: IDCAMS. The only valid values for SORTMESSAGELEVEL are 'ALL', 'NONE', or 'CRITICAL'.

Action: Contact ASG Customer Support.

#### DSS5780 "" IS INVALID MEMBER NAME

Explanation: IDCAMS. The dataset name coded does not conform to IBM requirements.

Action: Contact ASG Customer Support.

#### DSS5790 INVALID DATA FORMAT FOR SPECIFIED FIELD

Explanation: IDCAMS. The dataset name coded does not conform to IBM requirements.

Action: Contact ASG Customer Support.

#### DSS5795 DSNAMES MUST BE SAME FOR ALTER WITH MEMBER NAMES

Explanation: Related IBM message - IDC3530I. IDCAMS.

Action: The PDS names for 'ENTRY' and 'NEWNAME' must be the same for ALTER command when member names are specified. The occurrence of member names indicates a member name change within the same PDS.

#### DSS5800 MSG .... IS UNDEFINED IN JS8 ERROR TABLE

Explanation: IDCAMS. Internal diagnostic message.

Action: Contact ASG Customer Support because this may be release-dependent.

**DSS5808 INVALID GENERIC DATA SET NAME**

- Explanation: Related IBM message - IDC3203I. IDCAMS. The generic dataset name specified does not adhere to restrictions.
- Action: Contact ASG Customer Support.

**DSS5810 DSNAME FORMAT IS INVALID**

- Explanation: Related IBM message - IEF366I,IEF624I, IEF6501 or IEFC6501,IEF651I or IEFC651I
- Action: The DSN operand does not meet the requirements of temporary or permanent dataset naming conventions.

A temporary dataset name is one or two ampersands followed by one through nine alphanumeric or national characters, a hyphen, or a plus zero. The first character is alphabetic or national.

A permanent dataset is assigned either an unqualified or a qualified name.

An unqualified name is one through eight alphanumeric or national characters followed by a hyphen. The first character is alphabetic or national.

A qualified name consists of multiple names or nodes joined by periods. Each name or node is coded like an unqualified name of one through eight characters.

The maximum length of a qualified dataset name is 44 characters, including periods. For a GDG the maximum is 35 characters, including periods. For output tape the maximum is 17 characters, including periods.

**DSS5820 PATHENTRY DATA SET NOT FOUND**

- Explanation: Related IBM message - IDC3012I. IDCAMS.
- Action: The VSAM object specified in the PATHENTRY parameter could not be found in the catalog.

**DSS5821 RELATE ENTRY DATA SET NOT FOUND**

- Explanation: Related IBM message - IDC3012I. IDCAMS.
- Action: The VSAM cluster specified in the RELATE parameter could not be found in the catalog.

**DSS5825 DATA SET NOT FOUND IN CATALOG**

- Explanation: IDCAMS. The dataset specified in the IDCAMS statement could not be found in the catalog.
- Action: Contact ASG Customer Support.

**DSS5826 NO DATA SET NAMES FOUND IN CATALOG**

- Explanation: Related IBM message - IDC3012I. IDCAMS.
- Action: The catalog was searched for entries that matched the generic dataset name in the preceding control statement and none were found.

**DSS5830 RELATIVE GENERATION ALREADY EXISTS**

- Explanation: IDCAMS. An attempt was made to create a relative generation that already exists.
- Action: TB D

**DSS5834 DATA SET IS NOT SMS-MANAGED**

Explanation: Related IBM message - IDC3195I. IDCAMS.

Action: An Access Method Services ALTER command requested that a generation dataset (GDS) be rolled in, but the GDS is not managed by the storage management subsystem (SMS).

**DSS5835 DATA SET IS NOT A CANDIDATE FOR ROLLIN**

Explanation: Related IBM message - IDC3193I IDCAMS.

Action: An Access Method Services ALTER command request specified a ROLLIN operation for a non-VSAM dataset that is not in the deferred or rolled out state.

**DSS5840 TEMPORARY DSNNAME PREVIOUSLY DEFINED**

Explanation: A temporary DSN was coded twice in the same input stream.

Action: When it is later used as input, only the first occurrence will be used.

**DSS5850 GENERATION DOES NOT EXIST**

Explanation: The GDG referenced by the absolute or relative generation does not exist or it has not been cataloged.

Action: Contact ASG Customer Support.

**DSS5855 SPACE PARAMETER MISSING OR INCORRECT**

Explanation: IDCAMS. The space parameters TRACKS, CYLINDER, or RECORDS do not appear or are incorrectly specified.

Action: Contact ASG Customer Support.

**DSS5860 CATLG INVALID FOR TEMPORARY DATA SET**

Explanation: For a new temporary dataset, the second or third DISP parameter may not be KEEP or CATLG.

Action: This is because a temporary dataset is always deleted at the end of the job.

**DSS5870 KEEP INVALID FOR TEMPORARY DATA SET**

Explanation: Related IBM message - IEF648I

Action: For a new temporary dataset, the second or third DISP parameter may not be KEEP or CATLG. This is because a temporary dataset is always deleted at the end of the job.

**DSS5880 DATA SET ALREADY EXISTS ON VOLUME**

Explanation: IDCAMS. A new dataset being created on the target volume already exists on that volume.

Action: Contact ASG Customer Support.

**DSS5890 GENERATION DOES NOT EXIST ON VOLUME**

Explanation: IDCAMS. The catalog for this generation of a GDG refers to a DASD unit that does not contain the dataset.

Action: Contact ASG Customer Support.

**DSS5895 CANNOT CATALOG EXISTING GENERATION**

Explanation: IDCAMS. An attempt is being made to catalog a generation that is currently cataloged.

Action: Contact ASG Customer Support.

**DSS5910 DATA SET NOT FOUND ON DASD VOLUME**

Explanation: IDCAMS. The dataset referred to a volume that did not contain the dataset.

Action: Contact ASG Customer Support.

**DSS5920 DATA SET CANNOT BE FOUND**

Explanation: IDCAMS. The dataset cannot be found because it has not been passed, nor is it in the catalog.

Action: Contact ASG Customer Support.

**DSS5930 DATA SET IS ALREADY CATALOGED**

Explanation: Related IBM message - IDC3013I. IDCAMS. An attempt was made to catalog an already cataloged data set.

Action:

**DSS5931 WARNING - DATA SET IS ALREADY CATALOGED**

Explanation: Possible causes for this message are:  
- the file was created in an earlier step  
- the file was created by an earlier job but never deleted

Action: Contact ASG Customer Support.

**DSS5936 CATALOG CONTAINS NO GENERATIONS FOR THIS GDG**

Explanation: The generation data group contains no generation datasets.

Action: This message is issued when an IDCAMS statement specifies a DELETE command of a GDG base name with the "NONVSAM" parameter and there are no generations found in the catalog to delete.

**DSS5940 WARNING - ACTION WILL CAUSE POSSIBLE ERRORS**

Explanation: IDCAMS. This message refers to instances when an IDCAMS statement will execute but can produce undesirable or unpredictable results.

Action: Contact ASG Customer Support.

**DSS5941 Invalid volume serial specified for non-SMS dataset**

Explanation: Related IBM message - IDC3186I IDCAMS.

Action: '\*' is an invalid volume serial request for non-SMS managed datasets.

**DSS5942 UNABLE TO IDENTIFY UNIT NAME ""**

Explanation: IDCAMS. The value coded for the UNIT parameter or DEVT parameter cannot be identified as a valid device type for this installation.

Action: Contact ASG Customer Support.

#### DSS5943 UNIT "" IS AN INVALID SUBSYSTEM UNIT TYPE

Explanation: Related IBM message - IDC31604I IDCAMS.

Action: The specified unit is an invalid subsystem unit type for the SETCACHE command.

#### DSS5944 VOLUME "" INCONSISTENT WITH UNIT ""

Explanation: IDCAMS. The unit name specified by the UNIT or DEVT parameter is not valid for the volume serial number specified in the IDCAMS control statement.

Action: The device on which the volume is currently mounted is inconsistent with the device specified.

#### DSS5946 VOLUME "" CANNOT BE MOUNTED ON UNIT ""

Explanation: IDCAMS. The volume specified cannot be mounted because the unit address specified in the UNIT or DEVT parameter contains a permanently resident or reserved volume.

Action: Contact ASG Customer Support.

#### DSS5948 VOLUME "" WAS NOT PREVIOUSLY ALLOCATED

Explanation: IEHPROGM. No device was allocated for the required volume.

Action: Ensure that the volume serial number specified on the DD statement is the same as the volume serial number specified on the control statement.

#### DSS5950 GDG BASE NAME ALREADY EXISTS

Explanation: An IDCAMS DEFINE GENERATIONDATAGROUP statement was attempted and the entry already exists.

Action: Contact ASG Customer Support.

#### DSS5952 CATALOG INDICATES DATA SET MIGRATED/ARCHIVED

Explanation: IDCAMS. The dataset specified in the IDCAMS statement was found in the catalog; however, the volume serial indicates that the dataset has been migrated (archived) to another device.

Action: Contact ASG Customer Support.

#### DSS5955 GDG BASE ENTRY CONTAINS NON-VSAM ENTRIES

Explanation: IDCAMS. It is not valid to delete a GDG base entry that contains non-VSAM entries.

Action: The members of the GDG base must be deleted explicitly. Alternately, the GDG base and its members may be deleted with the use of the FORCE parameter in the DELETE command.

#### DSS5960 GDG BASE NAME DOES NOT EXIST

Explanation: An IDCAMS DELETE GENERATIONDATAGROUP statement was attempted, and the GDG was not cataloged.

Action: Contact ASG Customer Support.

#### DSS6010 NO OPERANDS FOUND

Explanation: Related IBM message - IEF605I or IEFC605I  
Action: The JCL statement being scanned does not contain any operand like JOB, EXEC, PROC, PEND, DD, or JES2/JES3 keywords.

#### DSS6020 OPERAND LONGER THAN ALLOWED CHARACTER SET

Explanation: The JCL statement being scanned contains an operand that is longer than eight characters.  
Action: Contact ASG Customer Support.

#### DSS6030 STATEMENT NAME TOO LONG OR MISSING BLANKS

Explanation: A name field was found next to the "/" but nothing else was found in the statement.  
Action: Contact ASG Customer Support.

#### DSS6040 STATEMENT MISSING POSITIONAL PARAMETER

Explanation: A DD statement did not contain the program keyword that is a required positional parameter.  
Action: This usually is caused by an omission or a misspelling.

#### DSS6042 STATEMENT HAS TOO MANY POSITIONAL PARAMETERS

Explanation: A JCL statement contains too many parameters. This could be caused by a misplaced parenthesis or apostrophe.  
Action: Contact ASG Customer Support.

#### DSS6050 UNKNOWN OPERATOR

Explanation: Related IBM message - IEF605I or IEFC605I  
Action: A JCL statement did not contain the operation field or a valid JCL verb (JOB, EXEC, DD, etc.) or operator command. This is usually the result of an omission or a misspelling.

#### DSS6060 UNEQUAL NUMBER OF LEFT PARENTHESES

Explanation: The end of the statement or the keyword was reached and the number of left parentheses exceeded the number of right parentheses.  
Action: Contact ASG Customer Support.

#### DSS6070 UNEQUAL NUMBER OF RIGHT PARENTHESES

Explanation: The end of the statement or the keyword was reached and the number of right parentheses exceeded the number of left parentheses.  
Action: Contact ASG Customer Support.

#### DSS6080 MISSING CLOSING QUOTE

Explanation: Related IBM message - IEF629I or IEFC629I  
Action: The statement above contains an opening quote but the closing quote is missing.

#### DSS6090 "" NOT DELIMITED BY EQUAL SIGN

Explanation: A parameter in the DD statement is not followed by an equal sign (=).

Action: Contact ASG Customer Support.

#### DSS6100 OPERAND EXTENDS BEYOND VALID CONTINUATION

Explanation: The statement above extends past column 72.

Action: Make sure that no parameters extend beyond column 72. If the statement needs to be continued, code '/' in columns 1 & 2 of the following statement and continue the interrupted portion of the statement in any column from 4 through 16. For example: // CONTINUATION CARD

#### DSS6105 OPERAND FIELD DOES NOT TERMINATE IN COMMA OR BLANK

Explanation: The statement above ends in column 71, is not a comma or a blank, and column 72 is non-blank, indicating a continuation.

Action: Contact ASG Customer Support.

#### DSS6110 INVALID CONTINUATION SYNTAX

Explanation: The statement above extends past column 72. Make sure that no parameters extend beyond column 72.

Action: If the statement needs to be continued, code '/' in columns 1 & 2 of the following statement and continue the interrupted portion of the statement in any column from 4 through 16. For example: // CONTINUATION CARD

#### DSS6125 UNBALANCED PARENTHESES IN THE " " FIELD

Explanation: Related IBM message - IEF622I

Action: An invalid delimiter character was detected. The error was the result of a missing parenthesis or an improperly coded separator.

#### DSS6130 CONTINUATION CARD STARTS PAST COLUMN 16

Explanation: The continuation line in this statement starts past column 16.

Action: The rule is that the first two positions must contain '/' followed by at least 1 but no more than 13 blanks before the first character of the continuation line.

#### DSS6135 ILLEGAL USE OF PARENS BEFORE DD KEYWORD

Explanation: An unbalanced or invalid use of parentheses was found.

Action: Contact ASG Customer Support.

#### DSS6140 EXPECTED CONTINUATION NOT RECEIVED

Explanation: Related IBM message - IEF618I or IEFC618I, IEF621I or IEFC621I

Action: The previous line of JCL implied that it should be continued on the next by the presence of a comma as the last character or a non-blank in column 72, but the next line contained the start of a new statement.

#### DSS6142 EXPECTED CONTINUATION NOT RECEIVED

Explanation: Related IBM message - IEF621I  
Action: The previous line of JCL implied that it should be continued on the next by the presence of a comma as the last character or a non-blank in column 72, but it was the last line in the JCL stream. There is no error when the JCL stream is submitted via the TSO SUBMIT command as a null JCL statement is inserted as the last line of the stream. If the stream is submitted through the INTERNAL READER, however, this will result in a JCL error.

#### DSS6144 STATEMENT ABOVE PROCESSED AS A CONTINUED COMMENT

Explanation: The statement could be a valid JCL statement but, because of a non-blank in column 72 in the previous line, it will be processed as a continuation of the comment field in the previous line.  
Action: Contact ASG Customer Support.

#### DSS6150 "" IS AN INVALID CHARACTER AS USED

Explanation: Related IBM message - IEF623I  
Action: The character enclosed in quotes is not valid as used in the statement. This is possibly caused by using a semicolon instead of a comma, parenthesis instead of a blank, or a plus sign instead of an equal sign (=).

#### DSS6151 INCORRECT USE OF APOSTROPHE ON "" STATEMENT

Explanation: Related IBM message - IEF629I  
Action: An incorrectly used apostrophe was found in the JCL statement. Single apostrophes are used to enclose certain parameters containing special characters or blanks. Two apostrophes within a parameter enclosed in apostrophes are used to represent an apostrophe.

#### DSS6152 IMPROPER USE OF PERIOD IN THE "" FIELD

Explanation: Related IBM message - IEF624I  
Action: A period was found in a parameter or field where a period is not permitted.

#### DSS6153 IMPROPER USE OF ASTERISK IN THE "" FIELD

Explanation: Related IBM message - IEF628I  
Action: An asterisk was found in a parameter or field where an asterisk is not permitted.

#### DSS6159 "==" IS AN INVALID DELIMITER

Explanation: Two equal signs in a row are not permitted.  
Action: Only one equal sign is needed after a keyword. Remove the unnecessary equal sign.

#### DSS6160 "" IS AN INVALID DELIMITER AS USED

Explanation: An invalid delimiter was encountered.  
Action: The only permissible values are space, equal sign, comma, left parens, and right parens.

#### DSS6161 "" IS AN INVALID DELIMITER AS USED IN AMP STRING



Explanation: An invalid delimiter was encountered.  
Action: The only permissible values are space, equal sign, comma, left parens, and right parens. These values are legal in certain places within the AMP subparameters.

#### DSS6162 EXCESSIVE LENGTH OF AMP STRING

Explanation: The string being scanned for the AMP parameter contains more than the maximum allowed subparameters.  
Action: Contact ASG Customer Support.

#### DSS6163 "" INVALID AS USED IN AMP STRING

Explanation: An invalid value was encountered during scan.  
Action: The value found to be in error is noted.

#### DSS6164 "=" NOT LEGAL AS USED IN AMP STRING

Explanation: An equal sign was encountered in an improper location in the AMP parameter. Check for "==" or following Trace/Amorg.  
Action: Contact ASG Customer Support.

#### DSS6165 "" INVALID WITH "" IN AMP STRING

Explanation: The first string is not legal as used in the second string.  
Action: Contact ASG Customer Support.

#### DSS6166 EMBEDDED BLANKS NOT LEGAL IN AMP STRING

Explanation: Blanks were found within the quoted string coded with the AMP parameter.  
Action: The blanks must be removed.

#### DSS6167 "" VALUE IS OUT OF RANGE - 0 TO ""

Explanation: For the subparameter noted, the numeric value coded is outside the permissible range as noted.  
Action: Contact ASG Customer Support.

#### DSS6168 "" PARAMETER REQUIRES SUBPARAMETERS

Explanation: The noted parameter has been coded without subparameters.  
Action: This parameter requires subparameters to be coded. See the IBM JCL Reference Manual for further details about parameters. This error could be caused by a missing equal (=) sign.

#### DSS6170 "" IS AN INVALID NAME AS USED

Explanation: The name referenced by the shorter message failed the standard JCL 8 character name edits.  
Action: The name could be a DD name, step name, PROC name, etc.

#### DSS7010 JES2/JES3 STATEMENT WAS UNIDENTIFIABLE

Explanation: The JES2/JES3 statement being scanned contained unidentifiable parameters.

Action: Contact ASG Customer Support.

#### DSS7020 FIRST OUTPUT FORM NUMBER CANNOT BE \*

Explanation: The JES2 /\* OUTPUT statement contained an \* for the value of the FORMS parameter.

Action: The FORMS parameter must have a numeric or alpha value as the first positional parameter.

#### DSS7030 MISSING VALUE FOR DEST

Explanation: The DEST keyword is coded in the JCL statement but the value is either missing or invalid.

Action: The DEST parameter is used to specify the location where the output is to be sent. Refer to the IBM JCL Reference Manual for proper coding instructions.

#### DSS7032 "" IS AN INVALID REMOTE DESTINATION

Explanation: The value coded for the DEST or NOTIFY parameter is not a valid destination for this system.

Action: Contact ASG Customer Support.

#### DSS7033 INVALID SYNTAX FOR DESTINATION

Explanation: DESTINATION parameter was coded incorrectly.

Action: This may be caused by too many commas or a missing subparameter.

#### DSS7034 "" IS AN INVALID NODE or USERID value

Explanation: The value coded for the NOTIFY subparameter of the OUTPUT statement is not a valid NODEID or valid USERID value(on this node).

Action: This value must be a valid definition known to TSO and/or JES. NOTE: This error is only detected if running JOB/SCAN APF authorized.

#### DSS7040 INVALID FORMS FIELD IN JES STATEMENT

Explanation: The value coded for the FORMS parameter must be alphanumeric and between 1 and 8 characters in length, indicating the form number to be used.

Action: Contact ASG Customer Support.

#### DSS7060 MISSING "=" IN JES STATEMENT

Explanation: A sub-parameter in the JES statement requires an operand and none was found.

Action: This could be because of missing parentheses or a comma coded instead of an equal sign.

#### DSS7070 "" IS INVALID FOR "" SUB-PARAMETER

Explanation: The operand found is not valid with the sub-parameter that it is coded with.

Action: See the JES manual for the sub-parameter noted to determine the problem.

#### DSS7071 "" IS AN INVALID PARAMETER FOR ""

Explanation: The sub-parameter found is not valid with the JES statement that it is coded with.

Action: See the JES manual for the JES statement noted to determine the problem.

**DSS7080 FIRST PARAMETER OF "FORMAT" MUST BE "PU" OR "PR"**

Explanation: The first parameter coded with the FORMAT command was not found to be "PU" or "PR". These are the only valid options as the first parameter to be coded with the FORMAT command.

Action: Be sure a comma follows the PU or PR separating it from DDNAME=.

**DSS7085 "PU" or "PR" must be in columns 11 and 12**

Explanation: The JES FORMAT PU/PR statement is column dependent.

Action: See the IBM JCL Reference Manual for the correct format.

**DSS7087 keyword VALUE MUST BE FROM 0000 TO nnnn (4 DIGITS)**

Explanation: The value for the named "keyword" is not in the required range of 0000 to "nnnn".

Action: Four digits must be coded. Correct and rescan the JCL.

**DSS7088 keyword hours/minutes MUST BE FROM 00 TO yy**

Explanation: The hours or minutes specified for the named "keyword" must be between 00 and "yy".

Action: Correct and rescan the JCL.

**DSS7089 keyword VALUE MUST BE FROM xx TO yy**

Explanation: The numeric operand for "keyword" is not within the required range of "xx" to "yy". Correct and rescan the JCL.

Action: Contact ASG Customer Support.

**DSS7099 FREQUENCY MUST BE A MULTIPLE OF 10; RANGE 10 TO 100.**

Explanation: The frequency that an operator WARNING message is to be issued must be specified as a multiple of 10 in the range 10 to 100.

Action: This number is a percentage used to calculate the amount of additional output between WARNING messages issued to the operator.

**DSS7100 VALUE MUST BE NUMERIC - FOUND " "**

Explanation: The operand found is not numeric.

Action: Check carefully that the value is numeric, with special attention to "O" (oh) and "0" (zero) and "L" and "1" (one).

**DSS7110 EXCESSIVE SUBPARAMETERS SPECIFIED FOR " "**

Explanation: The parameter was found to have more subparameters coded than allowed.

Action: Verify that there are not too many commas or a missing set of parentheses. Also check for missing operands.

**DSS7120 FIRST PARAMETER OF "NET" MUST BE "NETID" OR "ID"**

Explanation: The first parameter found in the NET command was not "NETID" or "ID". These are the only valid first parameters.

Action: Check for correct spelling or out-of-sequence parameters.

#### DSS7130 REQUIRED PARAMETER MISSING FOR " "

Explanation: A required keyword parameter is missing from the specified JES command.

Action: See the JES Reference Manual for the required parameters and/or check that all parameters are spelled correctly.

#### DSS7140 EXCESSIVE PARAMETER LENGTH " "

Explanation: The parameter found contains more characters than allowed.

Action: See the IBM JCL Reference Manual for parameter lengths and values. Check for misspellings and missing commas.

#### DSS7150 COMMAND " " IS INVALID IN JCL STREAM.

Explanation: The JES command specified cannot be submitted via JCL.

Action: This command is only valid from an operators' console.

#### DSS7160 SPECIFIED VALUE OUT OF RANGE FOR " "

Explanation: The value specified for the JES sub-parameter is outside the allowed range.

Action: Contact ASG Customer Support.

#### DSS7170 " " IS AN INVALID DELIMITER AS USED

Explanation: The delimiter noted is not valid in this position within this sub-parameter.

Action: Check for missing equals, commas, quotes or parens.

#### DSS7185 PRIORITY must immediately precede the JOB statement

Explanation: JES2 will ignore the /\*PRIORITY statement and flush the input stream until the next JOB statement or another /\*PRIORITY statement is found unless the /\*PRIORITY statement is immediately followed by a JOB statement.

Action: Contact ASG Customer Support.

#### DSS7186 Input stream flushed until next JOB statement

Explanation: JES2 will ignore the /\*PRIORITY statement and flush the input stream until the next JOB statement or another /\*PRIORITY statement is found unless the /\*PRIORITY statement is immediately followed by a JOB statement.

Action: Contact ASG Customer Support.

#### DSS7190 JES STATEMENT MUST NOT HAVE ANY PARAMETERS

Explanation: The JES statement coded must not have any parameters coded. Also, there may not be any comments coded in this statement.

Action: Contact ASG Customer Support.

#### DSS7200 JES SIGNON FORMAT INVALID, CHECK COLUMNS CAREFULLY

Explanation: The JES SIGNON statement is column dependent.

Action: All operands/parameters must start and end in the proper columns. For JES2: REMOTEnnn must begin in col 16 password1 must begin in col 25 password2 must begin in col 73 For JES3: workstatn must begin in col 16 A, if coded, is in col 22 B, if coded, is in col 23 password1 must begin in col 25 password2 must begin in col 35 password3 must begin in col 44. See the IBM JCL Reference Manual for more about format.

#### DSS7210 JES MODIFY "TRC" VALUE MUST BE 1, 2, OR 3

Explanation: The only allowed values for the JES MODIFY command are 1, 2, or 3.

Action: All other values are not legal and will cause a JCL error.

#### DSS7220 INTERNAL READER CONTROL STATEMENT FOUND

Explanation: An informational only message noting that an INTERNAL READER command was found within this job stream.

Action: Contact ASG Customer Support.

#### DSS7230 JES2 COMMAND FOUND WITHIN JES3 PROCESSING

Explanation: This job stream was being scanned with the JES3 processing option specified and a JES2 command was found.

Action: Contact ASG Customer Support.

#### DSS7240 EMBEDDED NULL (/ \* EOF) FOUND WITHIN JCL STREAM

Explanation: The / \* EOF statement has the same effect as a // null statement within a JCL stream.

Action: Contact ASG Customer Support.

#### DSS7250 FREQUENCY VALUE IS VALID ONLY WITH "WARNING" MESSAGE

Explanation: The FREQUENCY operand is only valid when the WARNING level of operator message is requested.

Action: Contact ASG Customer Support.

#### DSS7260 PARENTHESES NOT REQUIRED FOR " " PARAMETERS

Explanation: The noted sub-parameter operands should not be coded within parentheses and should be removed before running job stream.

Action: Contact ASG Customer Support.

#### DSS7265 PARENTHESES ARE REQUIRED FOR " " PARAMETERS

Explanation: The noted sub-parameter operands must be coded within parentheses and must be added before running the job stream.

Action: Contact ASG Customer Support.

**DSS7290 "IND" IS ONLY VALID 2ND PARAMETER WITH "\*" OR "ANY"**

Explanation: When "\*" or "ANY" is specified for the first value of SYSAFF, the only valid 2nd value is IND.  
 Action: Contact ASG Customer Support.

**DSS7300 "IND" MUST BE THE LAST PARAMETER FOR SYSAFF**

Explanation: The "IND" value for SYSAFF must be coded once only and must be the last value coded.  
 Action: Contact ASG Customer Support.

**DSS7310 " " 2ND VALUE NOT WARNING, CANCEL OR DUMP**

Explanation: The second value coded for the specified parameter is not 'WARNING', 'CANCEL' or 'DUMP'.  
 Action: These are the only permitted parameter values.

**DSS7320 JES3 CONTINUATION SHOULD START IN COLUMN 4.**

Explanation: When JES3 statements are continued, positions 1-3 must be "/\*" and position 4 should contain the first character of the next parameter. See the z/OS JCL manual, Chapter 3, Continuing JES3 Control Statements.  
 Action: Although JES3 does not flag continuations which start past column 4 IBM states that unpredictable results or error messages may occur if not following the IBM JCL Reference Manual rules for JES3 continuation.

**DSS7325 JES3 statement must be blank in cc-72.**

Explanation: When JES3 statements are not continued, position 72 must be blank. See the ESA JCL manual, Chapter 3, Continuing JES3 Control Statements.  
 Action: Contact ASG Customer Support.

**DSS7330 JES2 CONTINUATION IGNORED.**

Explanation: The JES2 continuation will be ignored. Refer to the JES2 Control Statements Section of your JCL Reference for rules on continuing a JES2 statement.  
 Action: Contact ASG Customer Support.

**DSS7600 CONTROL-M REQUESTS THAT THIS RUN BE CANCELED**

Explanation: The return code from Control-M indicates cancellation of the submission of the entire member.  
 Action: Contact ASG Customer Support.

**DSS7602 CONTROL-M ISSUED THE FOLLOWING MESSAGE(S):**

Explanation: The return code from CONTROL-M indicates cancellation of the submission of this job stream.  
 Action: A message from CONTROL-M was issued and is contained in message DSS7603 which follows.

**DSS7603 "text"**

Explanation: The text above is a message issued by CONTROL-M when a return code of 16 is returned by the Auto-Edit simulator (CTMJSP) or the CONTROL-M Exit 2 routine (CTMX002).

Action: This message should be preceded by message DSS7602.

#### DSS7604 CONTROL-M DETECTED PREMATURE END OF MEMBER

Explanation: The return code from CONTROL-M indicated that CONTROL-M still had some processing to do when End of Member was encountered.

Action: Contact ASG Customer Support.

#### DSS7610 UNIDENTIFIED POSITIONAL PARAMETER IN PATHDISP FIELD

Explanation: The PATHDISP parameter has certain permitted values.

Action: The positional parameter may be incorrect or misspelled.

#### DSS7611 INVALID CHARACTER IN THE PATH FIELD

Explanation: Related IBM message: IEF825I

Action: JOB/SCAN detected an invalid character in an HFS path.

#### DSS7612 REQUIRED PARAMETER MISSING IN THE " " FIELD

Explanation: Related IBM message: IEF646I

Action: JOB/SCAN did not find a required positional parameter or subparameter in the named field.

#### DSS7614 EXCESSIVE POSITIONAL PARAMETERS IN " "

Explanation: Related IBM message: IEF640I

Action: The system found too many positional parameters in the JCL statement. A misplaced comma, a duplication, or a null operand field could cause such an error. Note: After OS390R3, the COND statement requires additional parentheses if using the (0,EQ,SX.PROCY,EVEN) format. See the appropriate release of the IBM JCL Reference Manual.

#### DSS7615 UNIDENTIFIED PARAMETER IN THE " " FIELD

Explanation: The system did not recognize the value specified for the parameter.

Action: Contact ASG Customer Support.

#### DSS7616 IMPROPER SUBPARAMETER LIST IN THE PATHMODE FIELD

Explanation: The PATHMODE parameter contains an incorrect subparameter list for a positional parameter.

Action: Contact ASG Customer Support.

#### DSS7618 PATH NAME MISSING OR INVALID

Explanation: Related IBM messages - IEF817I, IEF825I

Action: The specified path name is missing or is spaces, contains embedded blanks or is not preceded by a slash.

#### DSS7619 CONFLICTING PATHOPT STATUS GROUP ENTRIES

Explanation: One or more values specified for PATHOPTS keyword found.

Action: For example PATHOPTS(OCREAT,OEXCL) for an existing file is not valid.

**DSS7620 PATH OR FILE NOT FOUND**

Explanation: The specified path name could not be found in any HFS file system.

Action: Contact ASG Customer Support.

**DSS7622 ONE OR MORE SUBDIRECTORIES NOT DEFINED.**

Explanation: In an HFS path, OCREAT was specified for a file, but one or more sub directories in the path name does not exist.

Action: Contact ASG Customer Support.

**DSS7623 HFS ERROR. RC=00000000 RSN=00000000**

Explanation: JOB/SCAN received an error from POSIX while validating an HFS path name.

Action: The Return Code and Reason Code returned by POSIX is displayed in hexadecimal.

**DSS7624 NOT AUTHORIZED TO VALIDATE PATH NAME**

Explanation: The user is not authorized to use z/OS UNIX System Services.

Action: Contact ASG Customer Support.

**DSS7625 PATH OR FILE ALREADY EXISTS.**

Explanation: An attempt was made create an HFS file or path but JOB/SCAN determined the path or file exists.

Action: Contact ASG Customer Support.

**DSS7650 " " CANNOT BE SPECIFIED WITHOUT " "**

Explanation: A JCL keyword was specified without one or more required accompanying keywords.

Action: Contact ASG Customer Support.

**DSS7660 INVALID VALUE FOR ROACCESS**

Explanation: Either of the ROACCESS parms were coded incorrectly.

**DSS7662 "DISALLOW" MUST APPEAR ALONE**

Explanation: When specifying DISALLOW, no other ROACCESS parms are allowed to be coded.

**DSS8000 xxxxxxxxx CANNOT BE USED AS MODEL DSCB**

Explanation: The model DSCB cannot refer to a tape file or a generation data group name. It must refer to a cataloged dasd dataset that has a valid VTOC entry.

Action: Refer to the IBM JCL Reference Manual, DD Statements, DCB Parameter for further information.

**DSS8010 INVALID DATA SET NAME SPECIFIED IN "LIKE" PARAMETER**



Explanation: The dataset name specified in the "LIKE" parameter must not be a temporary dataset (&&d-  
sname), partitioned dataset if a member name is included, or relative generation number for a  
GDG.

Action: The dataset must reside on a DASD device.

#### DSS8012 MODEL DATA SET IN "LIKE" HAS BEEN MIGRATED/ARCHIVED

Explanation: The dataset name specified in the "LIKE" parameter was found in the catalog but the volume  
serial indicates that it has been migrated (archived) to another device.

Action: Contact ASG Customer Support.

#### DSS8014 MODEL DATA SET SPECIFIED IN "LIKE" NOT FOUND

Explanation: Related IBM message - IGD16001I

Action: The dataset name specified in the "LIKE" parameter was not found. It is either not in the catalog  
or not on the volume specified by the catalog

#### DSS8020 INVALID SYNTAX FOR SUBSYS NAME

Explanation: The subsystem name must be one through four alphanumeric or national (\$,#,@) characters.

Action: The first must be alphabetic, \$, #, or @.

#### DSS8021 "" IS AN UNDEFINED MVS SUBSYSTEM NAME

Explanation: Related IBM message - IEF746I SUBSYSTEM XXX DOES NOT EXIST

Action: The subsystem name on the SUBSYS DD keyword is not defined to the system. SUBSYS must  
specify a valid MVS SUBSYSTEM name.

#### DSS8022 "" SUBSYSTEM DOES NOT SUPPORT SUBPARAMETERS

Explanation: The subsystem name on the SUBSYS DD keyword is defined to the system (in IEFSSxx) but was  
not defined to support subparameters, yet subparameters exist in the DD card.

Action: This message could also be the result of an inactive Subsystem defined in the IEFSSNxx mem-  
ber.

#### DSS8030 MISPLACED DD STATEMENT

Explanation: A DD statement has been found in an incorrect position in the JCL stream.

Action: A blank DD name cannot appear as the first JCL statement or immediately following a JCL state-  
ment other than another DD.

#### DSS8040 ZARA INITIALIZATION ERROR. RC = xxxx.

Explanation: ZARA initialization did not complete correctly. The return code from ZARA initialization attempt  
is shown.

Action: The ZARA subsystem is probably not up or is on the way down.

#### DSS8041 ZARA API MODULE ZARAAPI1 NOT AVAILABLE.

Explanation: The ZARA application program interface module is not available to JOB/SCAN.

Action: Check that the ZARAAPI1 program is in a library accessible to this run of JOB/SCAN.

#### DSS8042 ERROR ON ZARA READ FOR vvvvvv RC=xxxx / RSN=yyyy

Explanation: A READ request to ZARA for a volume sustained a return code and reason code not consistent with JOB/SCAN's expected results.

Action: The return code and reason code are displayed.

#### DSS8043 ZARA-SEVERE ERROR, FOR vvvvvv RC=xxxx / RSN=yyyy

Explanation: ZARA returned a severe error condition to JOB/SCAN.

Action: This could be caused by an error reading the database or a broken chain encountered.

#### DSS8044 ZARA-TERM HAS BEEN DONE. ZARA COMING DOWN.

Explanation: ZARA termination has been done. ZARA is coming down.

Action: Check with your installation for status of the ZARA subsystem.

#### DSS8045 ZARA SHOWS DATA SET NOT ON VOLUME vvvvvv.

Explanation: ZARA shows that the dataset specified is not on the volume specified.

Action: Correct your JCL to reflect correct volume or dataset.

#### DSS8046 ZARA SEQUENCE ERROR FOR VOLUME vvvvvv (#yyyy)

Explanation: ZARA shows that the sequence of volumes does not match the sequence specified in your JCL.

Action: Check for correct sequence of volume serials in the VOL=SER parameter of your JCL. #yyyy shows the ZARA sequence number.

#### DSS8047 ZARA TAPE vvvvvv OFF-SITE LOC IIIIIIII, SLOT sssssss

Explanation: ZARA shows that the tape specified is off-site at the shown location and in the shown slot.

Action: Check with your installation to retrieve the volume before running this JCL.

#### DSS8048 VOLUME vvvvvv IS NOT IN ZARA RANGE

Explanation: ZARA shows that the VOLSER specified is not in the range controlled by ZARA.

Action: To specify a tape not under ZARA control you must specify the EXPDT=98000 parameter.

#### DSS8049 VOLUME vvvvvv IS INACTIVE (SCR/DEL/INIT)

Explanation: ZARA shows that the VOLSER specified is inactive.

Action: Volume is in scratch/delete/init status.

#### DSS8050 ZARA API HAS ABENDED. SEE JLSSR FOR MSG.

Explanation: ZARA API (ZARAAPI1) has abended.

Action: The abend code is displayed in JLSSR output. ZARA option is turned off for any further processing.

#### DSS8051 D00YZARA BAD PARM LIST -VOLUME vvvvvv.

Explanation: D00YZARA (the API interface to ZARA) has passed an invalid parm list to ZARA.

Action: Contact ASG Customer Support. See also DSS0844 in the JLSSR dataset.

#### DSS8060 INVALID VALUE IN DEN PARAMETER

Explanation: The DEN parameter must be numeric and of the value, 1, 2, 3, 4, or 5.

Action: Contact ASG Customer Support.

#### DSS8080 MISSING SYSIN DD STATEMENT

Explanation: The PGM being executed requires a SYSIN DD statement and none was found.

Action: For more information, refer to the correct IBM utility manual for this program.

#### DSS8081 MISSING SYSPRINT DD STATEMENT

Explanation: The PGM being executed requires a SYSPRINT DD statement and none was found.

Action: For more information, refer to the correct IBM utility manual for this program.

#### DSS8082 MISSING SYSUT1 DD STATEMENT

Explanation: The PGM being executed requires a SYSUT1 DD statement and none was found.

Action: For more information, refer to the correct IBM utility manual for this program.

#### DSS8083 MISSING SYSUT2 DD STATEMENT

Explanation: The PGM being executed requires a SYSUT2 DD statement and none was found.

Action: For more information, refer to the correct IBM utility manual for this program.

#### DSS8084 MISSING SYSTSIN DD STATEMENT

Explanation: The PGM being executed requires a SYSTSIN DD statement and none was found.

Action: For more information, refer to the correct IBM utility manual for this program.

#### DSS8085 MISSING SYSTSPRT DD STATEMENT

Explanation: The PGM being executed requires a SYSTSPRT DD statement and none was found.

Action: For more information, refer to the correct IBM utility manual for this program.

#### DSS8086 USING DB2 PROGRAM NAME AS DEFAULT DB2 PLAN NAME

Explanation: The DB2 plan name was not specified, so DB2 uses the DB2 application program name as the default plan name.

Action: Contact ASG Customer Support.

#### DSS8087 USING TSOID+RUNLIB.LOAD AS DEFAULT DB2 LIBRARY NAME

Explanation: The DB2 library was not specified, so DB2 uses TSOID+RUNLIB.LOAD as the default library name.

Action: Contact ASG Customer Support.

#### DSS8088 DB2 "" INVALID

Explanation: A parameter specified in the DB2 control cards was invalid.

Action: Check the appropriate DB2 manual for the proper syntax.

#### DSS8089 INVALID DB2 KEYWORD: " "

Explanation: An invalid keyword was found in the DB2 control cards.

Action: Check the appropriate DB2 manual for the proper syntax.

#### DSS8090 DUPLICATE DB2 " " SPECIFIED

Explanation: An duplicate keyword was found in the DB2 control cards.

Action: Check the appropriate DB2 manual for the proper syntax.

#### DSS8091 DB2 PAREN - MISMATCHED OR MISPLACED: " "

Explanation: A mismatched or misplaced parenthesis was found in the DB2 control cards.

Action: Contact ASG Customer Support.

#### DSS8092 USING "I" AS THE DEFAULT FREE(DSN) MESSAGE FLAG

Explanation: The DB2 FREE FLAG was not specified.

Action: DB2 will use "I" as the default.

#### DSS8093 DB2 " " " " NOT SPECIFIED OR INVALID

Explanation: The referenced DB2 keyword was not specified or invalid.

Action: Check the appropriate DB2 manual for the proper syntax.

#### DSS8094 DB2 USING " " DEFAULT

Explanation: The referenced DB2 parameter was not specified. DB2 will use the default value.

#### DSS8095 DB2-" "

Explanation: The included text will indicate what error condition has been detected. Check the DB2 Command Reference for the proper syntax.

#### DSS8096 INVALID DB2 " " PARM SPECIFIED

Explanation: The referenced DB2 parm was invalid.

Action: Check the appropriate DB2 manual for the proper syntax.

#### DSS8097 DB2 BIND(DSN) PLAN NOT SPECIFIED. NO PLAN PRODUCED.

Explanation: Plan name was not specified in the DB2 BIND control cards. DB2 will process the BIND command but will not produce a plan.

#### DSS8098 NON-DB2 COMMAND(S) FOUND BUT NOT PARSED

Explanation: A non-DB2 command has been encountered within the DB2 control cards. JOB/SCAN performs error checking on the DB2 commands only.

#### DSS8099 ONLY IBM-SUPPLIED PLAN NAMES SHOULD BEGIN WITH "DSN"

Explanation: A plan name that begins with "DSN" has been found in the DB2 control cards. Only plan names supplied by IBM should start with "DSN".

#### DSS8100 INVALID VALUE FOR COPIES

Explanation: Related IBM message - IEF681I

Action: The valid values for the COPIES parameter are 1-255 in a JES2 system and 1-254 in a JES3 system. MVS uses the DD COPIES default of 1 when this parameter is coded incorrectly.

#### DSS8101 INVALID DB2 KEYWORD OR MISSING CONTINUATION CHAR

Explanation: An unrecognized DB2 keyword was specified or there was a continuation character missing from the end of the DB2 command line.

Action: Contact ASG Customer Support.

#### DSS8102 INVALID CONTINUATION ON PREVIOUS LINE

Explanation: There should not be a continuation character on the preceding DB2 command line.

Action: Contact ASG Customer Support.

#### DSS8103 MISMATCHED/MISPLACED APOSTROPHE IN DB2 PARM STRING

Explanation: An uneven number of apostrophes (single quotes) was specified in the DB2 parameter string.

Action: Apostrophes within the string are specified by two consecutive apostrophes.

#### DSS8104 DB2 ""

Explanation: A warning condition exists for this DB2 parameter.

Action: The parameter was invalid, not specified, or currently not found in the system. The message indicates the condition and any defaults that will be taken. Check the parameter & correct value if necessary.

#### DSS8105 DB2 DCLGEN TABLE NAME HAD MISSING OR MISMATCHED QUOTES

Explanation: Either one of two conditions was found: (1) the table name contained special characters (other than underscores) therefore the name must be enclosed by quotes; (2) An uneven number of single quotes was specified in the table name (quotes specified within the name are specified by two consecutive single quotes).

Action: Contact ASG Customer Support.

#### DSS8106 DB2 DCLGEN TABLE NAME - HYPHENS WILL REPLACE UNDERSCORES

Explanation: When the DCLGEN table name contains underscores and COBOL is the specified language, DCLGEN will translate the underscores to hyphens.

Action: Contact ASG Customer Support.

#### DSS8107 DB2 BIND MEMBER INVALID OR NOT FOUND: ""

Explanation: The DB2 BIND member name has an invalid syntax or cannot be found. DB2 first searches the BIND library(ies) then DBRMLIB DD libraries for the existence of this member.

Action: Contact ASG Customer Support.

#### DSS8108 DB2 BIND & DBRMLIB LIBRARIES NOT SPECIFIED OR INVALID

Explanation: The DB2 BIND member must exist in a library. This library can be specified in the BIND library parameter and/or on a DBRMLIB DD-card.

Action: This message indicates that there were no valid BIND library parameters and no valid DBRMLIB DD-card datasets.

#### DSS8109 DB2 BIND LIBRARIES NOT SEARCHED FOR MEMBERS

Explanation: Issued with error message DSS8108.

Action: No valid library was specified, so there was no search for the existence of the DB2 BIND member (s).

#### DSS8110 "" CONTAINS AN UNRESOLVED DDNAME REFERENCE

Explanation: Related IBM message - IEF686I A DD statement in the previous step or the last step of the job contains a DDNAME keyword parameter but the DD statement being referenced could not be found in that step.

Action: IF not corrected, DUMMY will be assumed. This error can also be caused if multiple DD statements contain DDNAME keywords that refer to the same DD statement. All DDNAME keywords must refer to unique DD statements.

#### DSS8111 DB2 COMMENTS CANNOT START IN COLUMN 1

Explanation: Comments within the DB2 Control Cards must begin with a "/" in column 2 or later.

Action: MVS recognizes the "/" in column 1 as end of the data stream.

#### DSS8112 DB2 RUN - EXTRANOUS PROGRAM INFO IGNORED

Explanation: The DB2 Run Program parameter contained more information than a single program name.

Action: DB2 accepts the first program name and issues an advisory message stating the additional information is ignored.

#### DSS8120 "" IS NOT A VALID "" PARAMETER

Explanation: The first parameter identified is not a valid value for the keyword identified in the second string enclosed in quotes.

Action: Contact ASG Customer Support.

#### DSS8125 TOTAL COPIES FOR OUTPUT STATEMENT WAS EXCEEDED - LIMIT IS 255

Explanation: The maximum copies of a sysout dataset that can be specified on an OUTPUT statement is 254 for JES2 and 255 for JES3.

Action: Contact ASG Customer Support.

#### DSS8130 OUTPUT STATEMENT MUST HAVE A NAME

Explanation: Each OUTPUT statement must have a name following the two slashes.

Action: Note that the OUTPUT parameter is valid only for MVS/XA users.

#### DSS8140 MBR NAME NOT ALLOWED WITH THIS DSNAME

Explanation: A member name cannot be specified for a VSAM sequential dataset or for a BDAM dataset.

Action: Contact ASG Customer Support.

#### DSS8150 DUPLICATE OUTPUT NAME

Explanation: Two OUTPUT statements have been encountered with the same name.

Action: Contact ASG Customer Support.

#### DSS8160 INVALID SYNTAX IN OUTPUT PARM

Explanation: OUTPUT parameter on the DD statement was coded incorrectly. OUTPUT=reference where reference can = \*.name  
\*.stepname.name \*.stepname.procstepname.name

Action: If more than one reference is listed, use parentheses and separate references with commas. A maximum of 128 references are allowed.

#### DSS8170 REFERENCED OUTPUT STATEMENT "" NOT FOUND.

Explanation: An OUTPUT statement was referenced, but could not be found.

Action: Check for correct spelling of the OUTPUT name and/or incorrect placement of it within the job stream.

#### DSS8180 NO CLASS SPECIFIED FOR SYSOUT STATEMENT

Explanation: No CLASS was specified on the SYSOUT statement or on the OUTPUT statement to which it applies.

Action: If CLASS is omitted on purpose, code a comma to note the omission.

#### DSS8190 "" CODED MORE THAN ONCE IN OUTPUT STATEMENT

Explanation: This parameter was found in the same output statement more than one time.

Action: Each parameter may be coded only one time per statement.

#### DSS8195 "" OUTPUT SUBPARM MUST PRECEDE FIRST "EXEC"

Explanation: Related IBM message - IEFC822I

Action: This keyword is not supported after the first exec statement.

#### DSS8200 PROC "" EXCLUDED AT USER REQUEST

Explanation: This PROC was not processed because it was excluded from processing and expansion by a Control Table. This was accomplished by:  
1) turning on the Run Option statement USE PROC SELECT TABLE and excluding the PROC from the AK Control Table

or

2) turning on the Run Option statement USE PROC EXCLUDE TABLE and including the PROC in the AN Control Table.

#### DSS8210 CANNOT ALLOCATE IMS LIBRARY

Explanation: Allocation failed for the IMS libraries containing the PSB's, DBD's or ACB's.

Action: Contact ASG Customer Support.

#### DSS8211 REFERENCE DDNAME IMS FOR PARM OF DLI

Explanation: When an IMS job is checked which has DLI as the first parm value, the dsn used to identify the DBD/PSB members is in the JCL with a DDNAME of IMS.

Action: This is an informational message displayed when another associated error is displayed.

#### DSS8212 REFERENCE DDNAME IMSACB FOR PARM OF DBB

Explanation: When an IMS job is checked which has DBB as the first parm value, the dsn used to identify the ACBLIB members is in the JCL with a DDNAME of IMSACB.

Action: This is an informational message displayed when another associated error is displayed.

#### DSS8213 REFERENCE ACBLIB IN PARMLIB FOR PARM OF BMP

Explanation: When an IMS job is checked which has BMP as the first parm value, the dsn used to identify the ACBLIB members is specified in the PARMLIB file. The DBD/PSB library specified in the JCL is only used by IMS for GSAM files. The ACBLIB which is actually used by IMS is in separate JCL for the control region which processes the database requests.

Action: This is an informational message displayed when another associated error is displayed.

#### DSS8220 CANNOT OPEN IMS LIBRARY

Explanation: Open failed for IMS libraries.

Action: Contact ASG Customer Support.

#### DSS8230 DD DEFINING IMS LIBRARY NAME MISSING OR INVALID

Explanation: The DD statement defining the IMS PSB, DBD, or ACB library is either missing or is invalid.

Action: Contact ASG Customer Support.

#### DSS8240 PSB "" NOT FOUND

Explanation: The PSB name was not found.

Action: Contact ASG Customer Support.

#### DSS8245 PSB "" HAS NO DBD NAMES

Explanation: The PSB contains no DBD entries.

Action: Contact ASG Customer Support.

#### DSS8250 DBD "" NOT FOUND



Explanation: The DBD referred to by the named PSB was not found.  
Action: Either the DBDLIB is not correctly paired to the PSB being used, or the DBD gen may have had errors. Contact your local IMS systems person.

#### DSS8251 FIND MACRO FAILED FOR DBD ""

Explanation: The FIND in the IMS interface failed for the named DBD. Although the member was found in the library, the DBD pointed to by the PSB did not contain a good DBD.  
Action: Contact local IMS Systems programmer for assistance.

#### DSS8260 ACB "" NOT FOUND IN DIRECTORY

Explanation: The ACB name was not found in the ACBLIB specified.  
Action: Either the library specified is not correct or the name of the ACB specified does not exist in that library.

#### DSS8261 FIND MACRO FAILED FOR PSB ""

Explanation: The FIND Macro failed when trying to point to the referenced PSB in the member read from ACBLIB.  
Action: Possible error could be an incorrect specification of a library, or an incorrect ACB member within the library specified. Contact the IMS systems programmer.

#### DSS8265 "" IS NOT A VALID IMS REGION TYPE

Explanation: The only valid IMS region types are BMP, DBB, DLI, IFP, MSG, RDI, UDR, and ULU.  
Action: Contact ASG Customer Support.

#### DSS8270 "" IS NOT A VALID xxx

Explanation: A module was loaded satisfactorily but an address referenced areas outside it. This module is probably not a valid PSB, DBD or ACB. Might also occur if the ACB/PSB/DBD loaded was not a valid IMS control block or the libraries specified are not matched to each other.  
Action: See the IMS systems programmer for assistance.

#### DSS8275 xxx "" COULD NOT BE LOADED

Explanation: A PSB or DBD could not be loaded from its respective library. This may indicate an invalid library, too small a region, or I/O error, etc.  
Action: The job log (JESLOG) may have CSVxxxx messages accompanying this error.

#### DSS8280 "" IS SPECIFIED MORE THAN ONCE IN IMS PARM

Explanation: A keyword parameter was specified more than one time in the parm field of the EXEC statement for an IMS step.  
Action: Contact ASG Customer Support.

#### DSS8285 "" PARAMETER IS LONGER THAN ALLOWED LENGTH

Explanation: The value specified for the noted parameter contains more characters than allowed.

Action: See the CICS Installation and Operations Guide for details, (SC33-0071, Chapter 5.6).

#### DSS8286 INVALID VALUE CODED - MUST BE 'A' OR 'C' OR 'P'

Explanation: The value specified for the LANG parameter is something other than the allowed set.

Action: The valid values are:  
A - Assembler  
C - Cobol.

#### DSS8287 INVALID VALUE CODED - VALUE MUST BE 'Y' OR 'N'

Explanation: The value specified for the noted parameter is something other than the allowed choice of 'Y' or 'N'.

Action: See the CICS Installation and Operations Guide for details, (SC33-0071, Chapter 5.6).

#### DSS8290 "" IS AN INVALID DELIMITER; "=" IS ONLY VALID VALUE

Explanation: A keyword was specified and not delimited with an equal (=) sign. (ie: PGM= or PSB=)

Action: Contact ASG Customer Support.

#### DSS8295 "" REQUIRED VALUE MISSING FROM PARM FIELD

Explanation: A keyword was not specified in the PARM field that is required for the DFHDRP program. PGM, PSB, and SSA are required parameters.

Action: See the CICS Installation and Operations Guide for details, (SC33-0071, Chapter 5.6).

#### DSS8300 DDNAME on the JES statement can not be found

Explanation: The DDNAME specified on the JES2 JOBPARM statement or the JES3 MAIN statement was not found.

Action: If specified, the JIJA libraries will be searched for PROC and INCLUDE members.

#### DSS8320 TMS INTERFACE "TMSSVC" MISSING

Explanation: Module TMSSVC could not be loaded.

Action: Concatenate the library containing the TMS load modules with the ASG load libraries.

#### DSS8330 TMS IS NOT ACTIVE

Explanation: TMS is not currently running in the system.

Action: Contact ASG Customer Support.

#### DSS8340 NO TMS RECORD FOR VOLUME-SERIAL xxxxxx

Explanation: A read of the TMS catalog was performed and no VOL=SER was found.

Action: Contact ASG Customer Support.

#### DSS8350 TMS SHOWS DATA SET NOT ON VOLUME xxxxxx

Explanation: The specified volume record was read and the dataset was not shown as existing on this volume, according to TMS.

Action: Contact ASG Customer Support.

#### DSS8352 TMS SEQUENCE ERROR FOR VOLUME xxxxxx (#nnnn)

Explanation: The volume sequence specified in the JCL or found in the catalog does not agree with TMS.

Action: The TMS sequence number is printed in parentheses for the volume specified.

#### DSS8360 TMS SHOWS SLOT NUMBER zzzz FOR VOLUME xxxxxx

Explanation: Since a slot number is recorded, it is possible that this volume is out-of-area and therefore not available for mounting.

Action: Contact ASG Customer Support.

#### DSS8370 TMS SHOWS TAPE xxxxxx OUT OF AREA "yyyy", SLOT zzzz

Explanation: There is a non-null value in the area code of the tape TMC record.

Action: If the OUT-OF-AREA shows LIBR or LIB, these are pseudo vaults and indicate that the tape is off-site (not available to mount). Enter out-of-area codes in the JOB/SCAN CONTROL TABLE "AM" (out of area codes) if you do not want the message. See [Control Table AM: Tape Management Locations \(Out-of-Area codes\)](#).

#### DSS8380 TLMS FAILED TO INITIALIZE, OPTION DISABLED

Explanation: Attempted to allocate and open the TLMS Volume Master File failed.

Action: Check for other message in the job log.

#### DSS8382 DATA SET NOT ON VOLUME xxxxxx

Explanation: The specified volume record was read and the dataset was not shown as existing on this volume, according to TLMS.

Action: Contact ASG Customer Support.

#### DSS8385 VOLUME xxxxxx IS NOT A TLMS VOLUME

Explanation: The specified volume is not in the volume ranges found in the Volume Master File.

Action: Contact ASG Customer Support.

#### DSS8395 VOLUME xxxxxx LOCATION IS yy

Explanation: The specified volume location is not in the data center "DC".

Action: Contact ASG Customer Support.

#### DSS8410 PECNTL DDNAME ALREADY ALLOCATED

Explanation: Attempted to allocate DDNAME "PECNTL" in order to verify program name under PANEXEC and it was already allocated.

Action: Contact ASG Customer Support.

**DSS8420 CANNOT ALLOCATE DDNAME PECNTL**

Explanation: Attempted to allocate DDNAME "PECNTL" in order to verify program name under PANEXEC and allocation failed.

Action: Contact ASG Customer Support.

**DSS8500 IF STATEMENT NOT COMPLETED - NO "THEN" FOUND**

Explanation: "IF" statement requires a "THEN" to identify the end of statement.

Action: Contact ASG Customer Support.

**DSS8505 MISPLACED "" STATEMENT**

Explanation: This statement cannot follow the "IF" or the "ELSE" that was previously coded or it cannot be associated with a previous procedure or step due to another error.

Action: Contact ASG Customer Support.

**DSS8510 UNEXPECTED END OF JCL - "IF" STMT WITHOUT MATCHING "ENDIF"**

Explanation: There must be the same number of "ENDIF" statements as there are "IF" statements.

Action: The end of the JCL was encountered before the correct number of "ENDIF"s were found.

**DSS8515 NUMERIC VALUE NOT VALID - RETURN CODE MUST BE 0-4095**

Explanation: The range for a numeric test on a return code must be 0 thru 4095.

Action: Contact ASG Customer Support.

**DSS8520 SYSTEM OR USER COMPLETION CODE FORMAT IS INVALID**

Explanation: System completion code format is Sxxx. User completion code format is Uxxx.

Action: Contact ASG Customer Support.

**DSS8525 UNBALANCED PARENTHESES IN THE "IF" STATEMENT**

Explanation: The number of right and left parentheses is not the same.

Action: Contact ASG Customer Support.

**DSS8530 INVALID SYNTAX IN "JCLLIB" STATEMENT**

Explanation: Code "ORDER=" before the dataset(s) on the JCLLIB statement.

Action: No other parameters are valid after the datasets. EX. //LIBS JCLLIB ORDER=(PROC.LIB1,PROC.LIB2)

**DSS8535 "JCLLIB" STATEMENT MUST BE CODED ONLY ONCE PER JOB**

Explanation: The JCLLIB statement must be coded only once in a job and must be coded before the first EXEC statement.

Action: Only the libraries on the first JCLLIB statement will be searched for PROCs and INCLUDEs.

**DSS8540 "JCLLIB" MUST BE CODED BEFORE THE FIRST "EXEC"**

Explanation: The JCLLIB statement must be coded before the first EXEC statement.

Action: Contact ASG Customer Support.

#### DSS8545 PROCLIB DD STMT CANNOT BE CODED WITH JCLLIB STMT

Explanation: The JCLLIB statement and the PROCLIB DD statement may not both be coded in the same job.

Action: The libraries that were coded on the first statement will be searched for PROCs and INCLUDEs.

#### DSS8550 RELATIONAL EXPRESSION INVALID IN "IF" STATEMENT

Explanation: The comparison is illogical.

Action: Contact ASG Customer Support.

#### DSS8555 ERROR IN "IF" STATEMENT

Explanation: The stated field has been coded incorrectly in the "IF" statement.

Action: Contact ASG Customer Support.

#### DSS8560 SYNTAX ERROR FOR "ABENDCC", "RC", "RUN", or "ABEND"

Explanation: The format for ABENDCC, RC, RUN, or ABEND has been coded incorrectly.

Action: Contact ASG Customer Support.

#### DSS8565 REFERBACK INVALID - " "

Explanation: The STEPNAME or PROCSTEPNAME specified for the ABENDCC, RC, RUN, or ABEND referback is incorrect.

Action: Contact ASG Customer Support.

#### DSS8570 MAXIMUM NUMBER OF NESTED INCLUDES - 15 MAXIMUM

Explanation: A maximum of 15 levels of nested INCLUDEs is allowed.

Action: Contact ASG Customer Support.

#### DSS8575 MAXIMUM NUMBER OF NESTED PROCs - 15 MAXIMUM

Explanation: A maximum of 15 levels of nested, cataloged, and instream PROCs is allowed.

Action: Contact ASG Customer Support.

#### DSS8580 "COMMAND" STMT MUST START & END WITH AN APOSTROPHE

Explanation: The first and last character of the "COMMAND" statement must be an apostrophe.

Action: Contact ASG Customer Support.

#### DSS8585 TOO MANY PARENTHESES IN THE " " PARAMETER

Explanation: Excessive parentheses coded on the parameter is invalid on MVS/ESA release 4.1 or greater.

Action: Contact ASG Customer Support.

#### DSS8590 NON-NUMERIC OR INVALID VALUE IN DPRTY PARAMETER

Explanation: Valid DPRTY parameters are 0 thru 15.

Action: Contact ASG Customer Support.

#### DSS8595 EXCESSIVE NUMBER OF VALUES SPECIFIED FOR DPRTY PARM

Explanation: No more than two values may be specified for the DPRTY parm.

Action: Contact ASG Customer Support.

#### DSS8600 INVALID SYNTAX FOR RELATIVE GDG DATA SET

Explanation: Only one set of parentheses are allowed around a relative gdg on MVS/ESA release 4.1 or greater.

Action: Contact ASG Customer Support.

#### DSS8605 INVALID SYNTAX FOR DATA SET MEMBER NAME

Explanation: The member name plus the parentheses may not exceed 10 characters on MVS/ESA release 4.1 or greater.

Action: Contact ASG Customer Support.

#### DSS8610 OUTPUT KEYWORD CODED IN AN EARLIER DD STATEMENT

Explanation: Related IBM message - IEF811I, DUPLICATE VERB & LABEL WITHIN A STEP

Action: The SYSOUT DD statement should not be overridden with DDNAME=, if both the SYSOUT DD and the DD statement referred to by DDNAME= contain the OUTPUT= keyword.

```
//OUT1 OUTPUT DEST=DSSI02 //PROC1 PROC
//STEP1 EXEC PGM=IEFBR14 //DD1 DD SYSOUT=H,OUTPUT=*.OUT1 //PEND1 PEND
//CALL EXEC PROC1
//DD1 DD DDNAME=DD2 //OUT2 OUTPUT DEST=DSSI02
//DD2 DD SYSOUT=H,OUTPUT=*.OUT2
```

#### DSS8615 MAXIMUM NUMBER OF 15 LIBRARIES MAY BE SPECIFIED

Explanation: A maximum number of 15 libraries may be specified for the JCLLIB statement.

Action: Contact ASG Customer Support.

#### DSS8620 NESTED IF MAXIMUM EXCEEDED

Explanation: A maximum number of 15 levels of nested IF/THEN/ELSE/ENDIF constructs can be coded.

Action: Contact ASG Customer Support.

#### DSS8900 DSN = " "

Explanation: The Data Set Name referred to in the previous message.

Action: Contact ASG Customer Support.

#### DSS8919 SYMLIST KEYWORD REQUIRED FOR EXPORT

Explanation: An EXPORT statement was found but a corresponding SYMLIST keyword was found to be missing.

Action: Add the required SYMLIST keyword to the EXPORT statement.

DSS8940 TEMPORARY DATA SET WILL BE USED.

Explanation: An unnamed dataset has been detected which will cause a temporary dataset to be allocated to this DD name.

# Return Codes

This topic contains explanations of the various product return codes that might appear in JOB/SCAN and/or DOCU/TEXT execution messages.

## Batch Execution of JOB/SCAN

See the table.

Code	Meaning
0	No error or warning messages were issued.
4	At least one warning message was issued, but no error messages were issued.
8	At least one error message was issued.
12	At least one severe error message was issued.
16	At least one critical error message was issued.
20	A fatal error message was issued. No scan was performed, at least for one member.

---



# ASG-JOB/SCAN Plug-in for Eclipse IDE

This documentation introduces the features and benefits of ASG-JOB/SCAN Plug-in, explains how to install, and execute the product in Eclipse environment.

It consist of following topics:

- [Introduction](#)
- [Installation](#)
- [User Interface](#)
- [Connecting to z/OS and ASG-JOB/SCAN](#)
- [Setting the JCL Scan Options](#)
- [Setting the Processing Options](#)
- [Working with z/OS Datasets](#)
- [Working with Local Files](#)
- [Validating JCL](#)
- [Reports](#)
- [Markers](#)
- [Keyboard Shortcuts](#)
- [ASG-JOB/SCAN REST API Documentation](#)
- [Uninstallation](#)

# Introduction

## What is ASG-JOB/SCAN Plug-in?

ASG-JOB/SCAN Plug-in is a Java plug-in that strategically integrates the mainframe capabilities of ASG-JOB/SCAN into the Eclipse environment. The Plug-in provides developers and production control staff an intuitive user interface to initiate JCL integrity, verification, and reporting. The Plug-in supports the key functions implemented in ASG-JOB/SCAN.



The JOB/SCAN Plug-in is a separately licensed option of ASG-JOB/SCAN. You need a license key to access it. Contact [ASG Customer Support](#) to obtain a license.

## Features of JOB/SCAN Plug-in:

- Offers a modern and easy-to-use User Interface (UI) to interact with the z/OS system.
- Evaluates JCL for any syntax errors.
- Enforces site-specific JCL standards.
- Submits jobs to the mainframe for execution.
- Provides effective reporting facility.
- Includes RESTful web services API to automate ASG-JOB/SCAN scans in a DevOps toolchain environment.

This document assumes that the user:

- has installed ASG-JOB/SCAN and running on a supported z/OS system.
- has authorization to access resources on the z/OS system.
- is familiar with the functions implemented in ASG-JOB/SCAN. If you are new to ASG-JOB/SCAN, refer to [ASG-JOB/SCAN® 8.0.0 Documentation](#).

# Installation

This topic contains instructions for installing ASG-JOB/SCAN Plug-in. These instructions include product requirements, installing the product, and verifying the installation.

## Installation Requirements

Before you install JOB/SCAN Plug-in, verify that the following requirements are met:

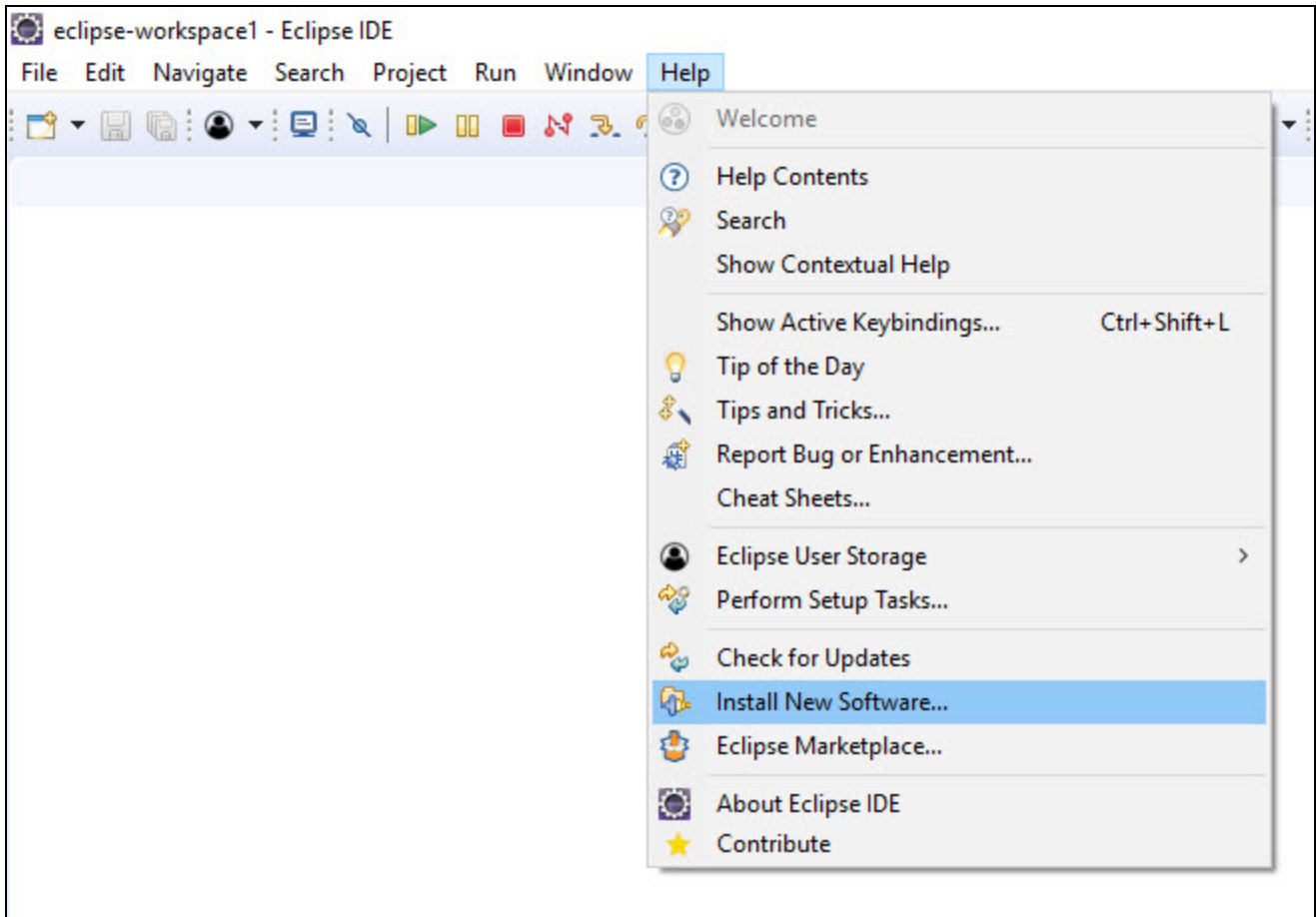
- ASG-JOB/SCAN Version 8.0.0 must be installed in the z/OS system. This topic does not explain the installation procedure for ASG-JOB/SCAN. See [ASG-JOB/SCAN Installation](#) for information on installing ASG-JOB/SCAN.
- Latest monthly cumulative service for ASG-JOB/SCAN must be applied. Visit ACCESS Portal: <https://access.asg.com> to verify any monthly maintenance apply to this product release.
- Install and Configure IBM Liberty Profile Server to enable ASG-JOB/SCAN REST Services. For details on installing Liberty server see the [Installer Reference Guide](#) section for guides on installing the ASG-JOB/SCAN REST Services. For more information about Liberty Servers z/OS installation requirements see [IBM Installing Liberty](#).
- User must have a TSO user ID defined to the mainframe system and should be able to access z/OS Unix System Services with home directory setup.
- User's TSO ID must be able to access and use ASG-JOB/SCAN under ISPF.
- Eclipse IDE version 2018-06 or higher must be installed. Eclipse IDE is an open-source software and it can be downloaded at no charge from <https://www.eclipse.org/downloads/>
- Java Runtime Environment (JRE) must be installed.
  - You can install JRE from OpenJ9: <https://www.eclipse.org/openj9/>
  - You can install JRE from Oracle: <https://www.java.com/en/download/manual.jsp>
- IBM z/OS® Management Facility (z/OSMF) Version 2.4 must be installed and running before using ASG-JOB/SCAN Plug-in. User must have read and write access to z/OSMF. To obtain read or write access, contact your system administrator. To verify if z/OSMF is configured as per ASG-JOB/SCAN Plug-in requirements, follow the procedure in [Verifying the z/OSMF Configuration](#).

## Installing ASG-JOB/SCAN Plug-in

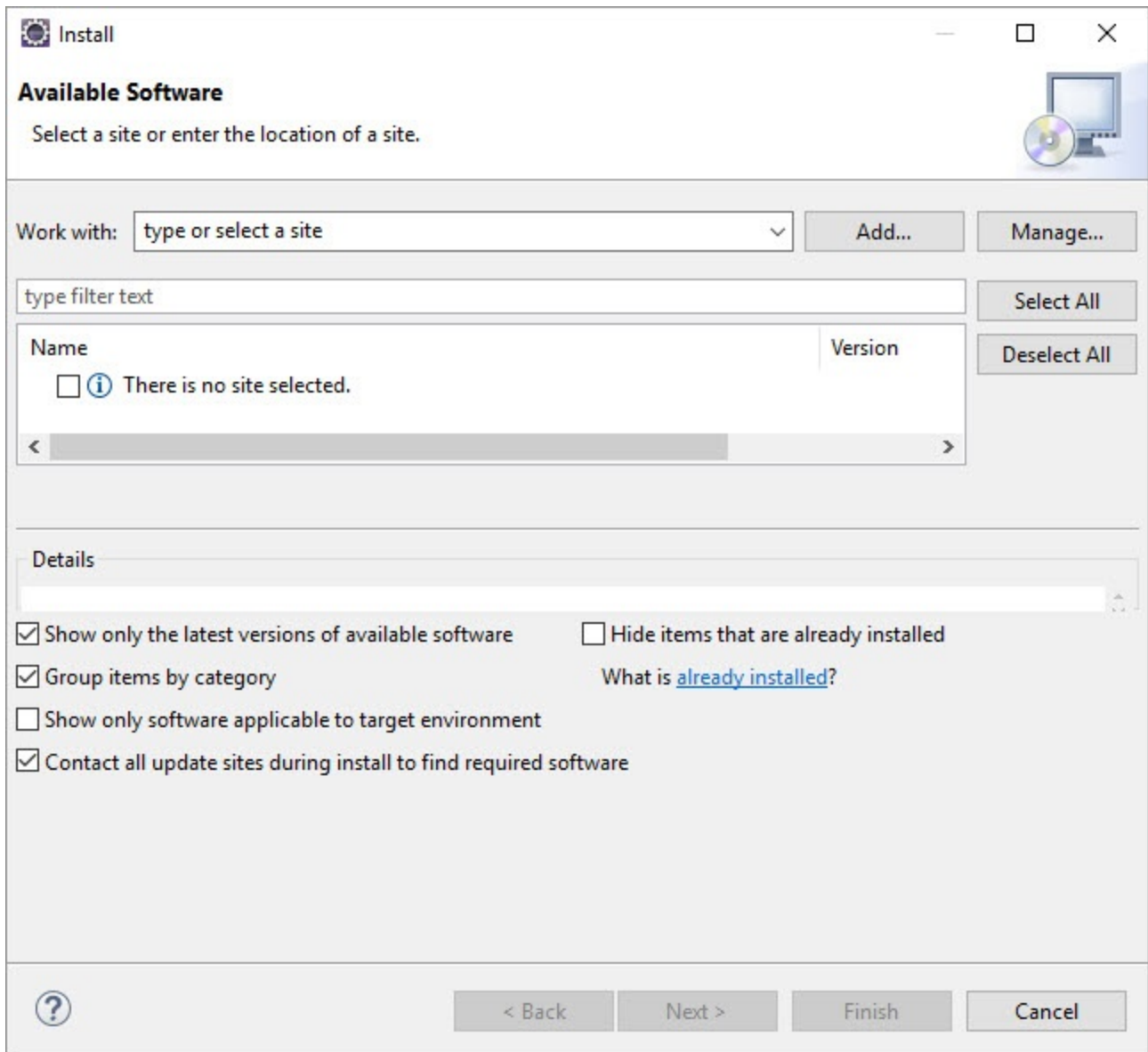
Follow these instructions to install the ASG-JOB/SCAN Plug-in after you download the product files.

### *To install ASG-JOB/SCAN Plug-in:*

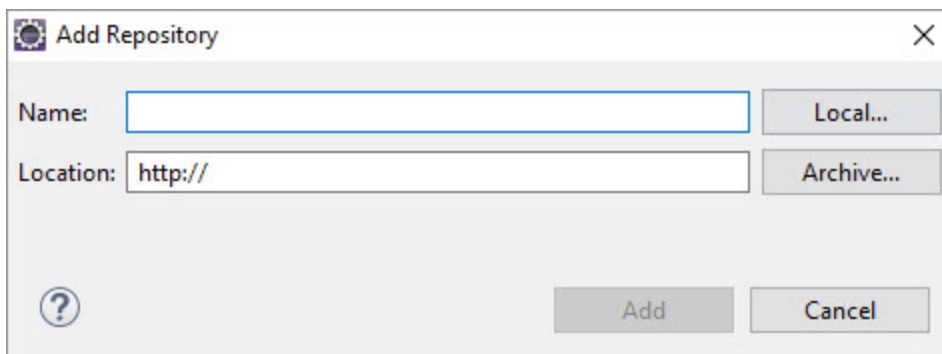
1. Start Eclipse.
2. From the menu, select **Help >Install New Software...**



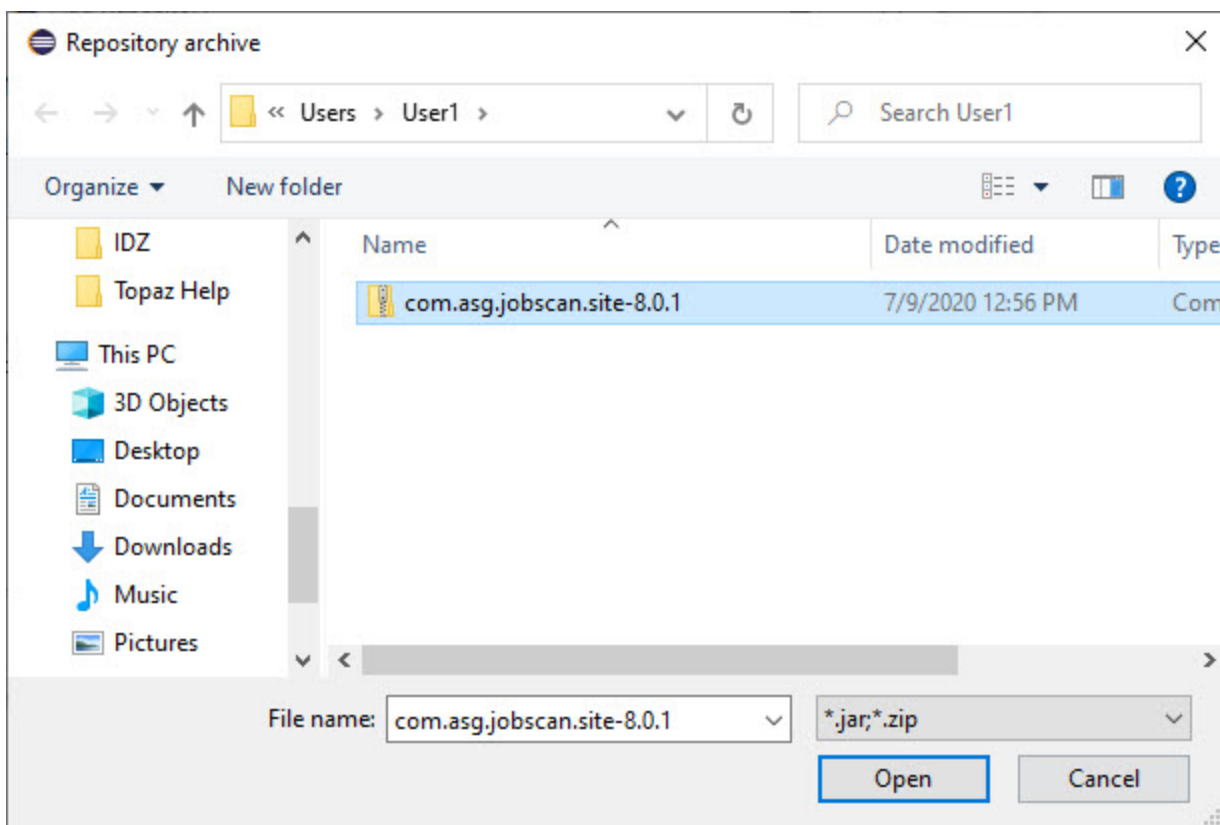
3. In the **Install** dialog, click **Add...**



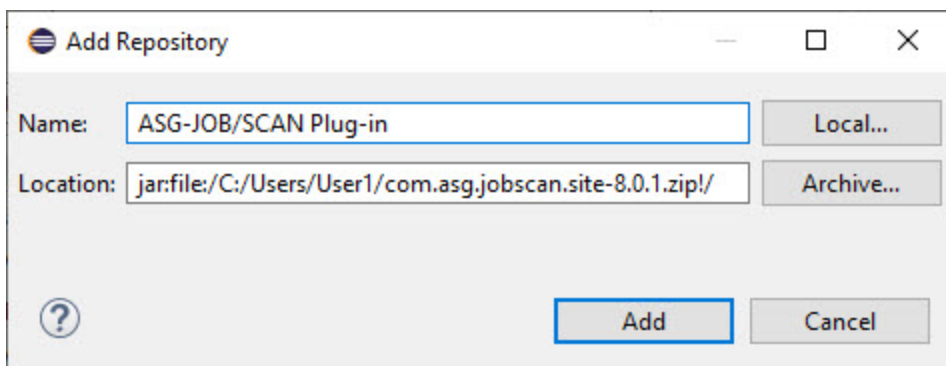
4. In the **Add Repository** dialog, enter a **Name** for the repository and click **Archive...**



5. In the **Repository archive** dialog, browse to select the ZIP file containing the install files, and click **Open**.

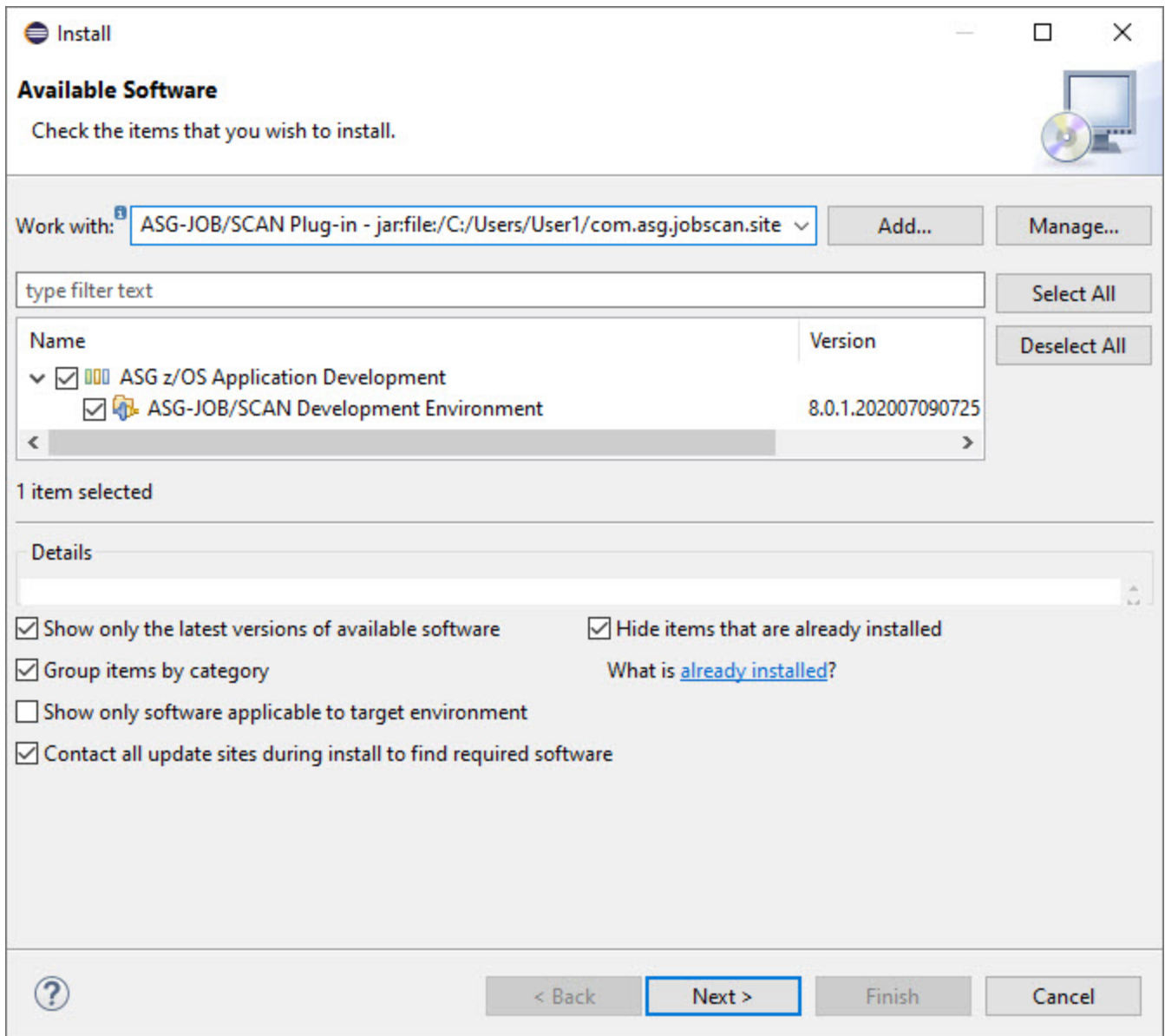


6. In the **Add Repository** dialog, click **Add**.

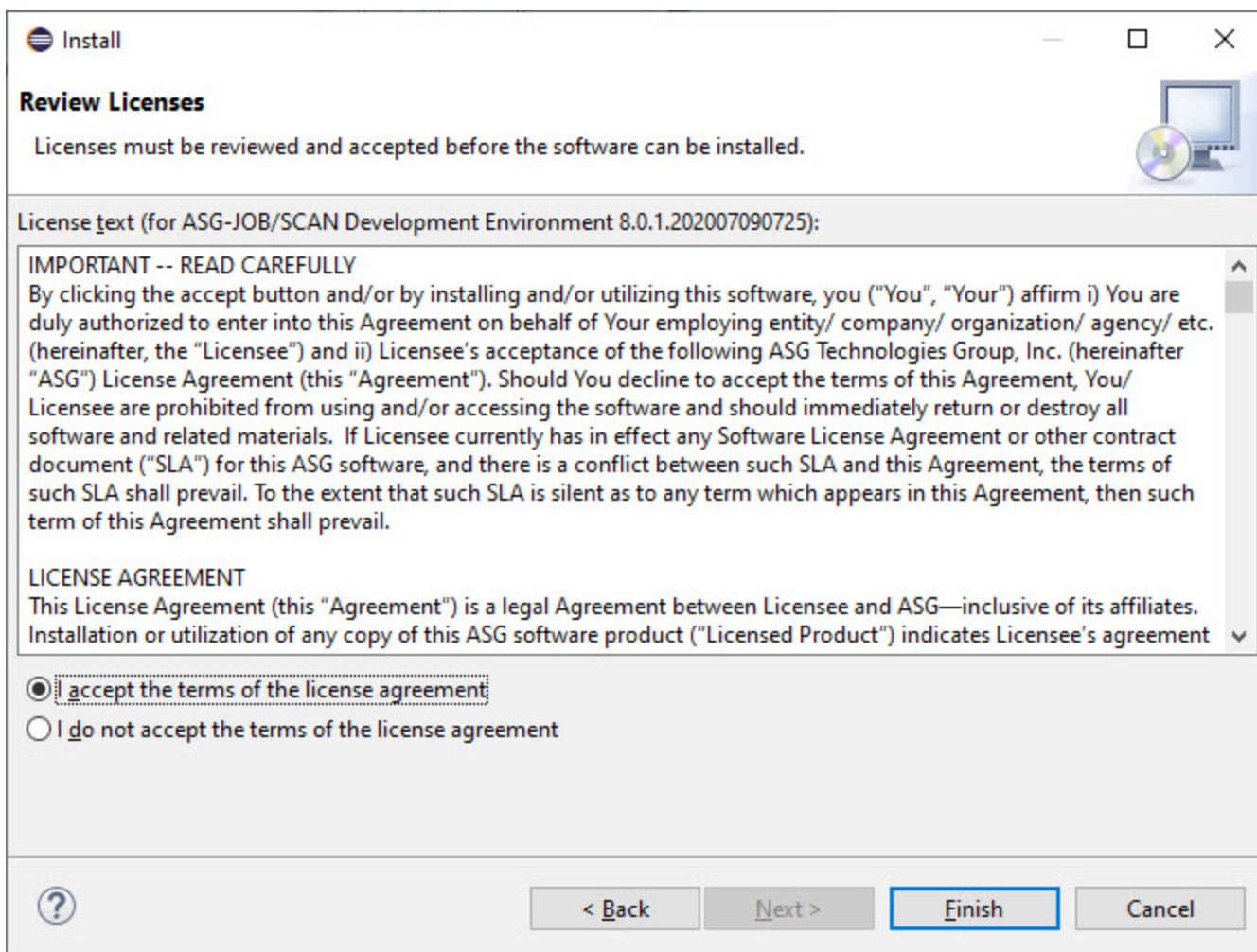


The **ASG z/OS Application Development** category appears in the **Name** area.

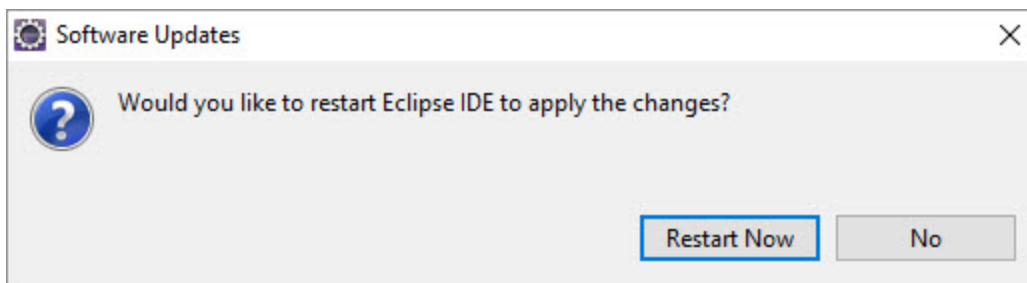
7. Select the **ASG z/OS Application Development** category and make sure that **ASG-JOB/SCAN Development Environment** feature is selected.
8. Click **Next**.



9. Review the install items and click **Next**.
10. Review the license and select **I accept the terms of license agreement**.



11. Click **Finish**. You may receive a security warning. Click **Install anyway** to continue with the installation.
12. When the installation is complete, you are prompted to restart the Eclipse application. Click **Restart Now**.



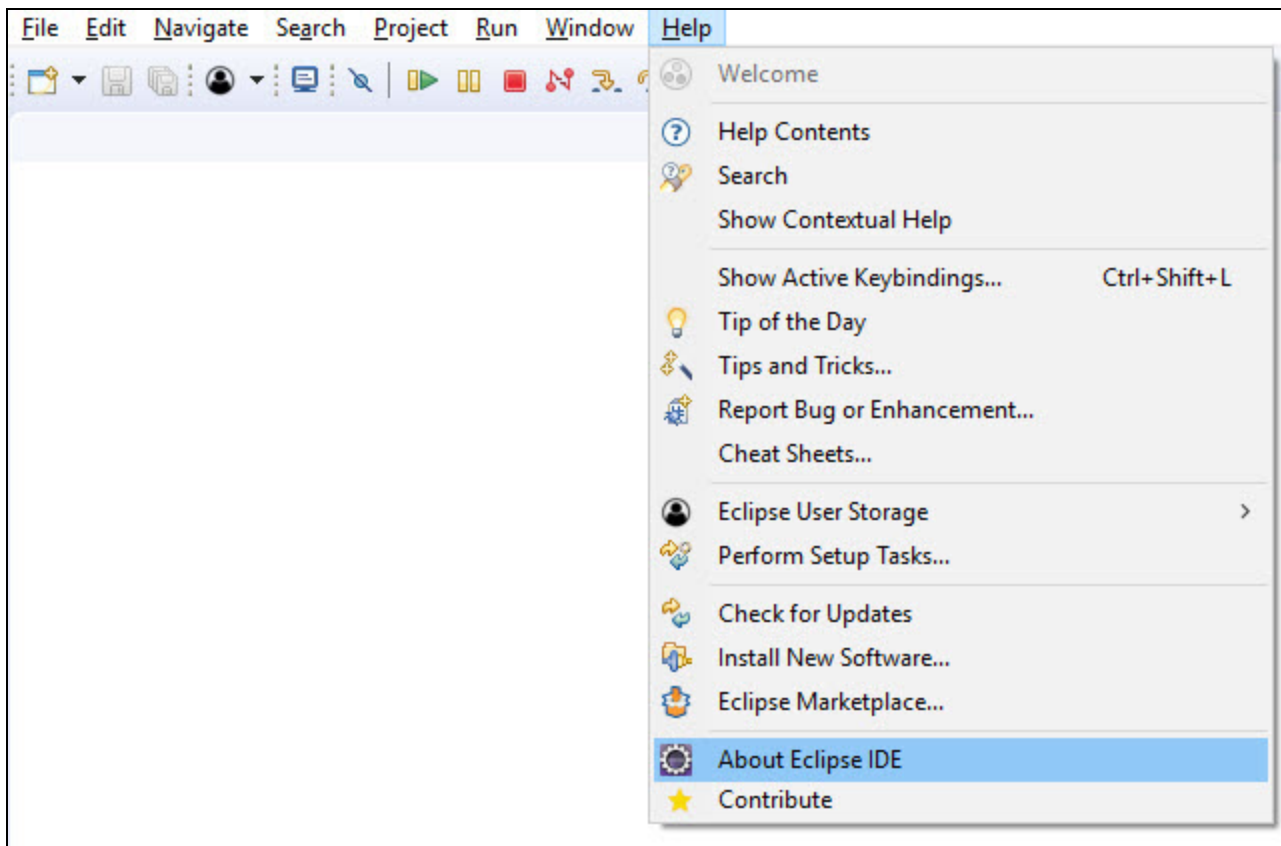
## Verifying the Installation

Follow the instructions in this section to verify whether the ASG-JOB/SCAN Plug-in is installed successfully or to find whether the plug-in is already installed.

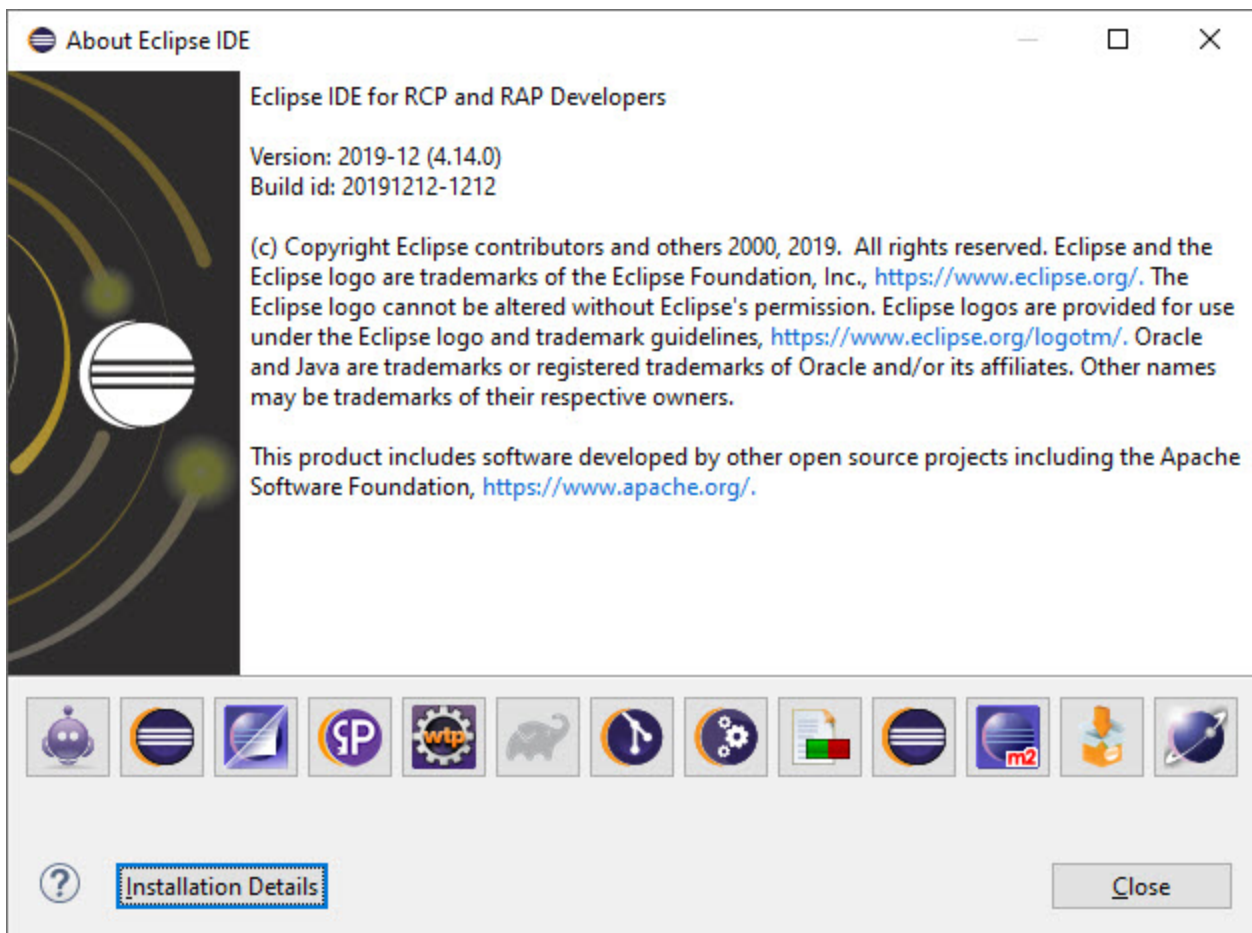


**To verify that the JOB/SCAN Plug-in is installed:**

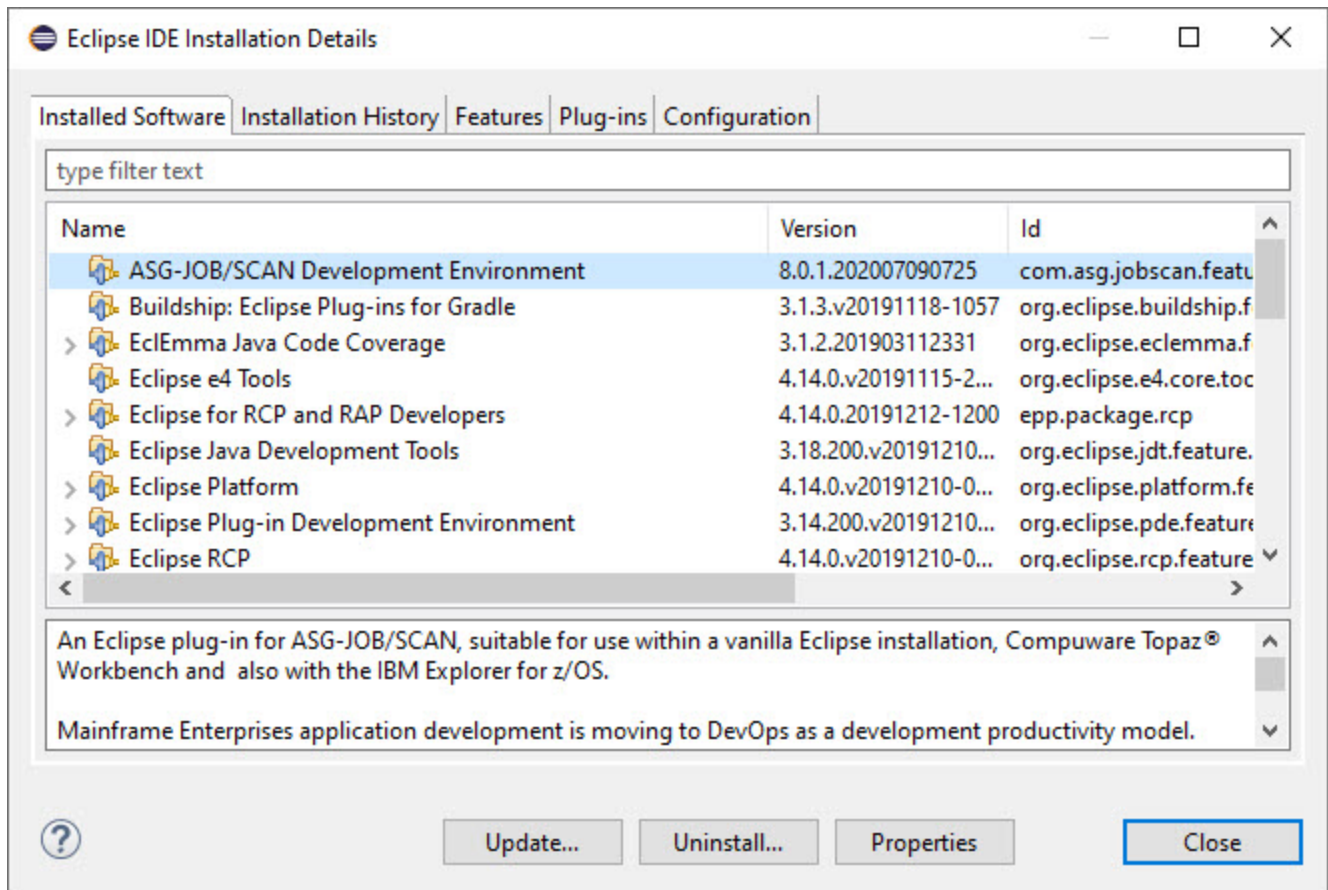
1. Start Eclipse.
2. From the menu, select **Help > About Eclipse IDE**.



3. In the **About Eclipse IDE** dialog, click **Installation Details**.



4. In the **Installed Software** tab, you should see **ASG-JOB/SCAN Development Environment** in the list of installed software. This indicates that JOB/SCAN Plug-in has been installed.



You can check the **Installation History** tab (Help > About Eclipse IDE > Installation Details > Installation History) for information on when a feature is installed.

## Verifying the z/OSMF Configuration

IBM z/OS® Management Facility (z/OSMF) must be installed and configured before using ASG-JOB/SCAN Plug-in. This topic provides instructions to verify if z/OSMF is configured as per ASG-JOB/SCAN Plug-in requirements.



This section does not explain the z/OSMF configuration procedure. See [Configuring z/OSMF](#) in IBM Knowledge Center for more information.

### Step to verify z/OSMF Configuration

- Verify if z/OSMF server is running

Issue the console command `/D A, IZU*` in the command line and check if the started tasks jobs, **IZUANG1** and **IZUSVR1**, are running. If the jobs are not active, issue `/S IZUANG1` and `/S IZUSVR1` commands to activate.

- Verify if z/OSMF REST services are enabled

Open a web browser to z/OS system (host name and port) and add the following REST call to the URL: **GET** /zosmf/restjobs/jobs. You should see the list of the jobs that are submitted from your user ID.

- Verify if z/OSMF data set and file REST services are enabled

See if the job **IZURFSEC** (available in the sample jobs package) is submitted and completes with the return code '0000'. To verify if services are running, issue a REST service request. See [List the z/OS data sets](#) in IBM Knowledge Center for syntax on how to issue a request.

The job IZUSVR1 shows the actual REST services that are running in z/OSMF. JOB/SCAN Plug-in requires both restjobs and restfiles services. To make sure that these services are available, check the output (STDOUT) files of IZUSRV1 job, and see if it contains a message that looks like:

```
CWWKT0016I: Web application available (default_host): https://<your-zosmf-
url>:<zosmf-port>/zosmf/<service>/
```

where <service> is a REST service name.

For more assistance work with your administrator.

- Verify access to z/OSMF
  1. In a browser, open the URL: `https://<hostname>:<port>/zosmf/`

Where:

hostname is the hostname or IP address of the system in which z/OSMF is installed

port is the secure port for the z/OSMF configuration

2. Enter the z/OS USER ID that was used to configure z/OSMF
3. Enter the z/OS PASSWORD associated with the z/OS user ID.
4. Click LOGIN.

The Welcome page of IBM z/OS Management Facility is displayed. This confirms that you are authenticated to access z/OSMF.

## References:

- What setup is needed for z/OSMF - [https://www.ibm.com/support/knowledgecenter/SSLTBW\\_2.3.0/com.ibm.zos.v2r3.izua300/IZUHPINFO\\_WhatSetupsNeeded.htm](https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.izua300/IZUHPINFO_WhatSetupsNeeded.htm)
- Software prerequisites for z/OSMF - [https://www.ibm.com/support/knowledgecenter/en/SSLTBW\\_2.2.0/com.ibm.zos.v2r2.izua300/IZUHPINFO\\_SoftwarePrereqs.htm](https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.2.0/com.ibm.zos.v2r2.izua300/IZUHPINFO_SoftwarePrereqs.htm)
- Security configuration requirements for z/OSMF - [https://www.ibm.com/support/knowledgecenter/SSLTBW\\_2.3.0/com.ibm.zos.v2r3.izua300/izuconfig\\_SecurityStructuresForZosmf.htm](https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.izua300/izuconfig_SecurityStructuresForZosmf.htm)
- Configuring z/OSMF - [https://www.ibm.com/support/knowledgecenter/SSLTBW\\_2.3.0/com.ibm.zos.v2r3.izua300/IZUHPINFO\\_PartConfiguring.htm](https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.izua300/IZUHPINFO_PartConfiguring.htm)

## Verifying the Liberty Server Configuration

IBM Liberty Server for z/OS must be installed and a Liberty server for ASG-JOB/SCAN REST Services must be configured before using ASG-JOB/SCAN Plug-in. This topic provides instructions to verify if the Liberty server for ASG-JOB/SCAN REST Services is ready as per the ASG-JOB/SCAN Plug-in requirements.



This section does not explain the Liberty Server for ASG-JOB/SCAN REST Services configuration procedure. See [Installer Reference Guide](#) for more information.

A Liberty server instance may be started as a z/OS started task or started from the z/OS UNIX shell. To operate Liberty servers as z/OS started tasks, BBGZSRV by default would be the started task name.

When a Liberty server seeks to use z/OS authorized services, BBGZANGL by default is required. ASG-JOB/SCAN REST Services does not require a Liberty angel process to be setup.



For more information on the various options for configuring and starting a Liberty runtime environment on the z/OS, see the [IBM WebSphere Application Server for z/OS Liberty](#).

### Step to Verify Liberty Server for ASG-JOB/SCAN REST Services

- Verify if Liberty server is running

Issue the console command `/D A, BBGZSRV*` in the command line and check if one or more started tasks **BBGZSRV\*** are running. If you are not using **BBGZSRV\*** sample proc, substitute this name with the name of the “JCL Start proc name” dedicated to ASG-JOB/SCAN REST Services.

- Start a Liberty server

If started task **BBGZSRV\*** or the name of the “JCL Start proc name” dedicated to ASG-JOB/SCAN REST Services is not running you need to start it. The following example shows the command that you would use to start a Liberty server configured for ASG-JOB/SCAN REST Services named “jjsrest” using the sample bbgzsrv proc:

```
START BBGZSRV.JJSREST, PARMS='jjsrest'
```

The following example shows the command that you would use to stop a ASG-JOB/SCAN REST Server (note: z/OS START command syntax of “S membername[.identifier]”, was used to identify jjsrest as the name identifying the task to be started, so you can issues commands against this task)

```
P JJSREST
```

- Review the messages of the Liberty server. If ASG-JOB/SCAN REST Services were successfully started these messages would be logged to the output STDOUT DD.

The following message verifies that JOB/SCAN REST Services is up and running:

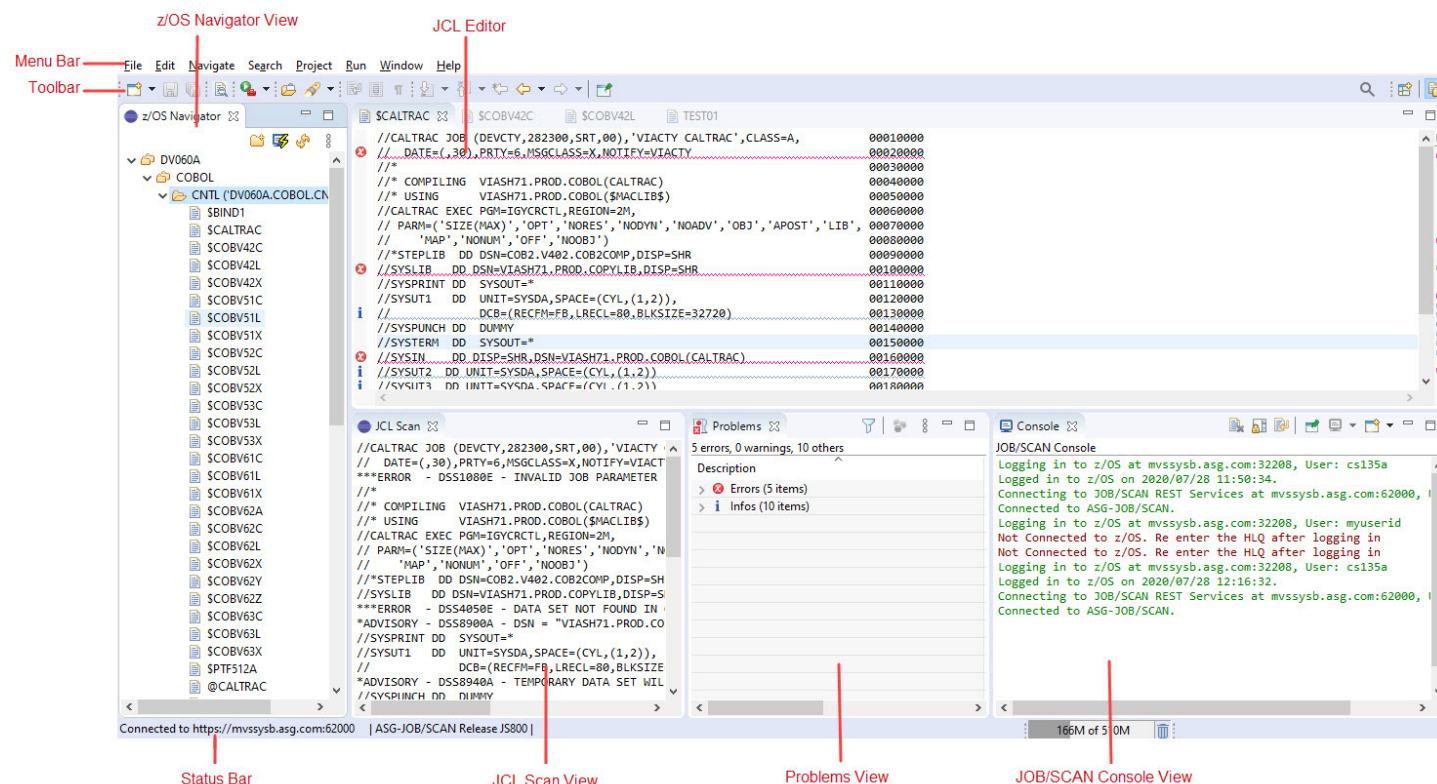
```
CWWKF0011I: The jjsrest server is ready to run a smarter planet.
```

**Sample STDOUT DD Output**

```
Launching jjsrest(WAS FOR Z/OS 19.0.0.12/wlp-1.0.35.cl191220191120-0300) on IBM J9 VM, version
8.0.6.0 - pmz6480sr6-20191107_01(SR6) (en_US)
[AUDIT   ] CWWKE0001I: The server jjsrest has been launched.
[AUDIT   ] CWWKZ0058I: Monitoring dropins for applications.
[AUDIT   ] CWWKT0016I: Web application available (default_host):
http://mvssyse.asg.com:49080/api/docs/
[AUDIT   ] CWWKT0016I: Web application available (default_host):
http://mvssyse.asg.com:49080/api/explorer/
[AUDIT   ] CWWKT0016I: Web application available (default_host):
http://mvssyse.asg.com:49080/jobscan/
[AUDIT   ] CWWKZ0001I: Application jobscan started in 0.556 seconds.
[AUDIT   ] CWWKF0015I: The server has the following interim fixes active in the runtime:
PH19989,PH19528. For a full listing of installed fixes run: productInfo version --ifixes
[AUDIT   ] CWWKF0012I: The server installed the following features: [distributedMap-1.0, el-3.0,
jaxrs-2.1, jaxrsClient-2.1, jndi-1.0, jsonp-1.1, jsp-2.3, openapi-3.0, servlet-4.0, ssl-1.0,
zosSecurity-1.0].
[AUDIT   ] CWWKF0011I: The jjsrest server is ready to run a smarter planet. The jjsrest server
started in 3.358 seconds.
[AUDIT   ] CWPKI0820A: The default keystore has been created using the 'keystore_password'
environment variable.
```

# User Interface

This topic describes about the commonly used interface elements that are required when using the JOB/SCAN Plug-in functions. It also provides information on how to view these elements when not displayed in the workbench.



## Menu Bar

The **Menu Bar** contains menus and menu options that are used to perform a task. JOB/SCAN commands are displayed as additional options in the Edit menu. However, you must be connected to JOB/SCAN's server before running the JOB/SCAN commands. Some options in the menu bar are available only when a file is active in the JCL Editor.




## Toolbar

The **Toolbar** provides quick access to tasks that are commonly performed by a user. JOB/SCAN commands are displayed as buttons on the toolbar. However, you must be connected to JOB/SCAN's server before running the JOB/SCAN commands. Some options on the toolbar are available only when a file is active in the JCL Editor.

## z/OS Navigator View

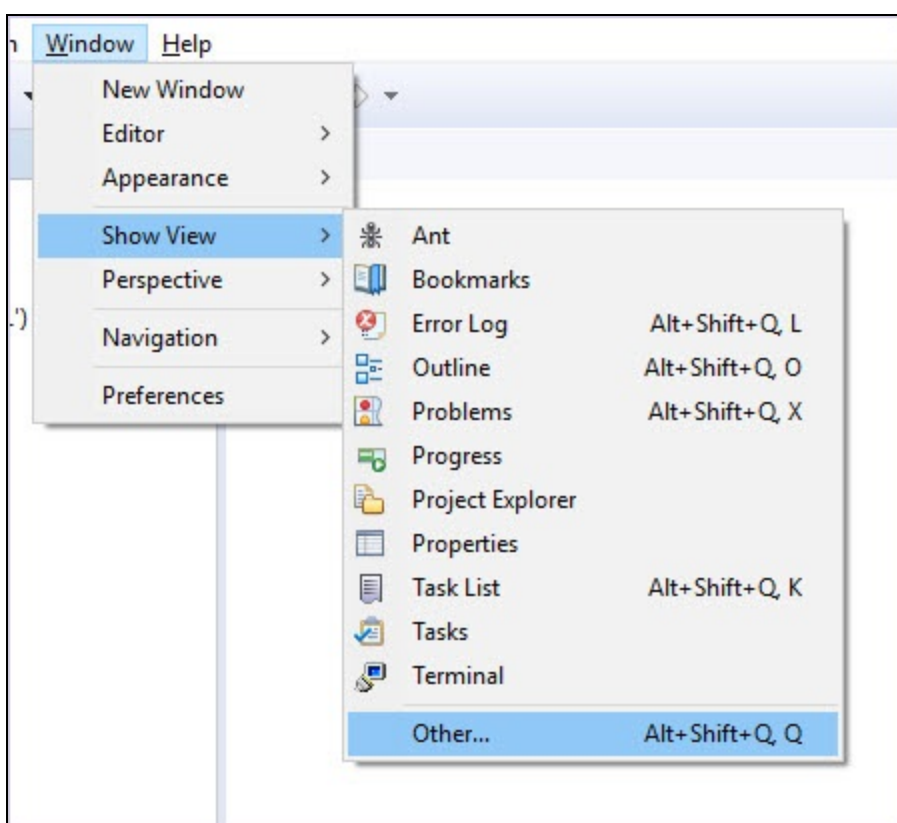
**z/OS Navigator** view displays datasets that are fetched from z/OS. The datasets are displayed under a High-Level Qualifier (HLQ) in a tree-like structure. For more information about displaying datasets, refer to [Displaying Datasets](#).

z/OS Navigator View consists of following icons:

Icon	Name	Description
	Add High Level Qualifier	Fetches datasets from the z/OS system and displays under the specified HLQ. See <a href="#">Working with z/OS Datasets</a> .
	Connect to z/OS	Establishes connection to z/OS system. See <a href="#">Connecting to z/OS and ASG-JOB/SCAN</a> .
	Refresh View	Refreshes the list of datasets displayed. See <a href="#">Refreshing Datasets</a> .

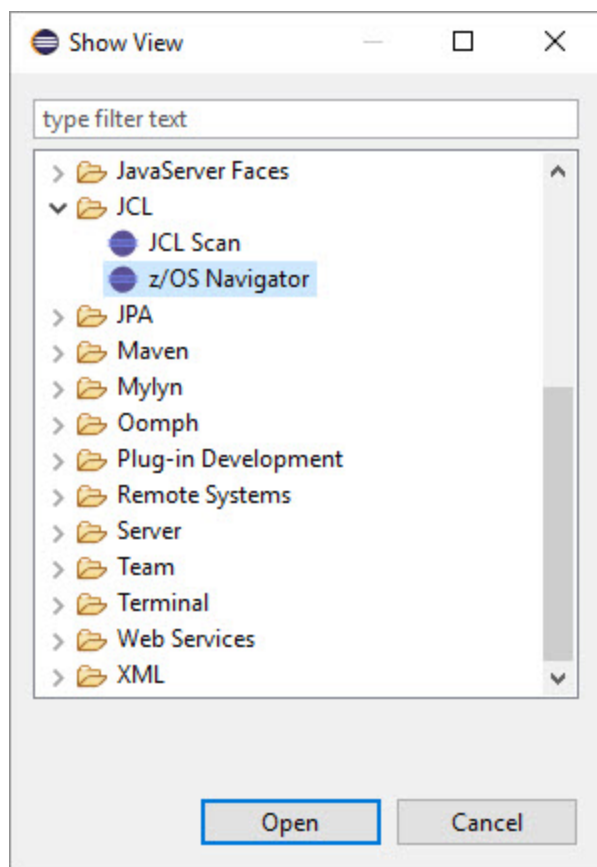
### To display z/OS Navigator view:

1. From the menu, select **Window > Show View > Other....**



2. In the **Show View** dialog, do one of the following:
  - Enter **z/OS Navigator** in the **type filter text** box.
  - Locate and expand the **JCL** folder.





3. Double-click on **z/OS Navigator** or select **z/OS Navigator** and click **Open**.

## Status Bar

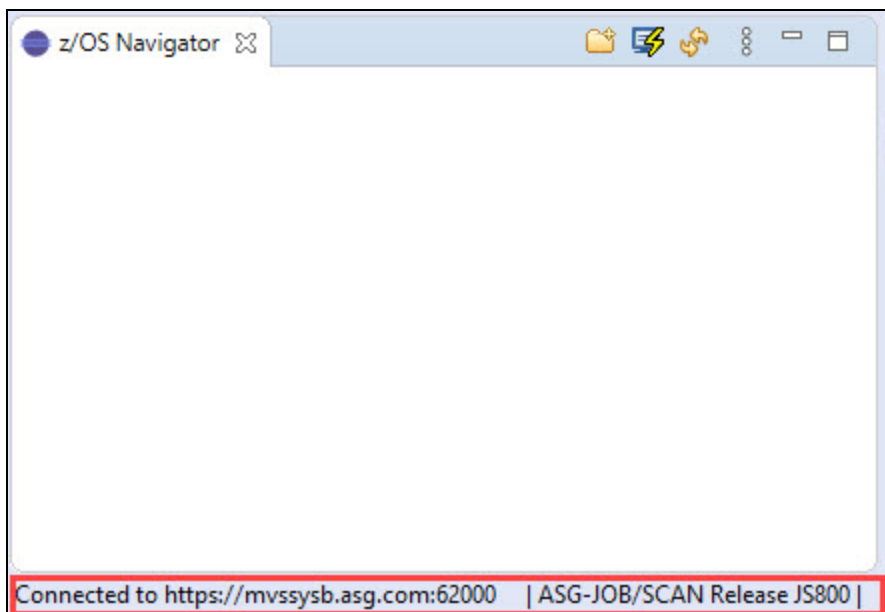
The **Status Bar** displays the z/OS connection status.



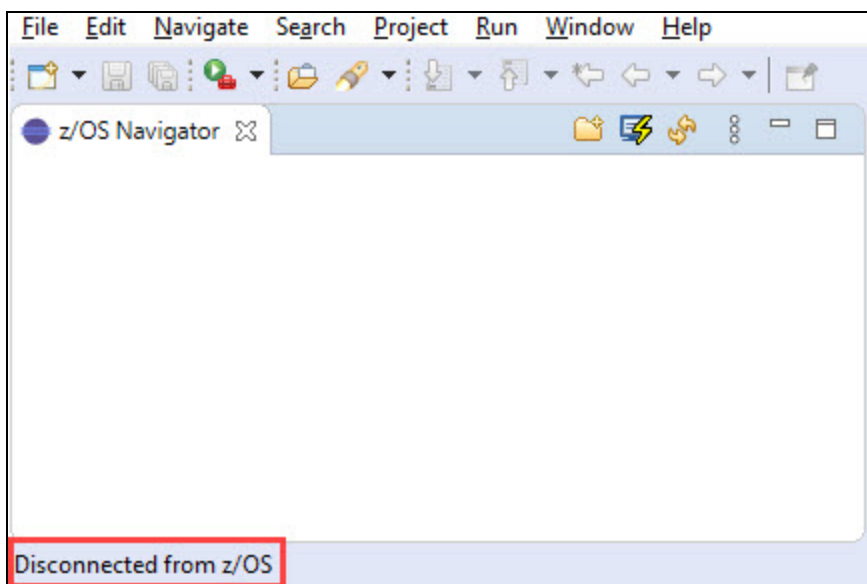
The z/OS connection status is shown only when the **z/OS Navigator** view is active.

After establishing the connection to z/OS and JOB/SCAN server, the status bar displays the following message:

**Connected to** <host address or IP address of the z/OS system>:<port number of the host machine on which the JOB/SCAN REST API service is running> | **ASG-JOB/SCAN Release JS**<version of the ASG-JOB/SCAN it is connected> |



When the connection is lost or when the session is disconnected from z/OS, the status bar display the message: **Disconnected from z/OS**.



## JCL Editor

The **JCL Editor** is typically used to view, edit, save and validate a JCL file. You can view more than one file in the editor area, but only one can be active at a time. Tabs in the editor area indicate the names of files that are currently open for editing. Some options in the menu bar and toolbar are available only when a file is active in the JCL Editor.

```

//COBV51X JOB (DEVCTY,282300,SRT,00),'COBOL 5.1',CLASS=A,MSGCLASS=X, 00010000
//.....NOTIFY=&SYSUID,TIME=(0,10),PRTY=6,REGION=4096K 00020000
//* 00030000
//RUNCOB PROC HLQ=DV060A.COBV51,MEM=NC101 00040000
//RUNPGM EXEC PGM=&MEM 00050000
//STEPLIB DD DSN=&HLQ..LOADPDSE,DISP=SHR 00060000
//* DD DSN=COB2.R40.COB2LIB,DISP=SHR 00070000
//*RINT DD SYSOUT=* 00080000
//PRINT DD DSN=&&PGMOUT,UNIT=SYSALLDA, 00090000
//.....DISP=(NEW,PASS),SPACE=(CYL,(1,1)) 00100000
//* 00110000
//***** OUTPUT COPY STEP ***** 00120000
//COPYOUT EXEC PGM=VIAGENER,REGION=2M,COND=(8,LE,RUNPGM) 00130000
//STEPLIB DD DSN=ESW.PROD.LOAD,DISP=SHR 00140000
//SYSIN DD DUMMY 00150000
//SYSPRINT DD DUMMY 00160000
//SYSUT1 DD DSN=&&PGMOUT,DISP=(OLD,DELETE) 00170000
//SYSUT2 DD DSN=&HLQ..OUTPUT(&MEM),DISP=SHR 00180000
//* 00190000
//***** OUTPUT TIMESTAMP STEP ***** 00200000
//TOUCHOUT EXEC PGM=VIATOUCH,PARM='MEMBER=&MEM.,LIBTYPE=OUTPUT', 00210000
//.....REGION=2M,COND=(8,LE,RUNPGM) 00220000
//STEPLIB DD DSN=ESW.PROD.LOAD,DISP=SHR 00230000
//SYSPRINT DD DUMMY 00240000
//SYSLIB DD DSN=&HLQ..OUTPUT,DISP=SHR 00250000
//* 00260000
//.....PEND 00270000
//* 00280000

```

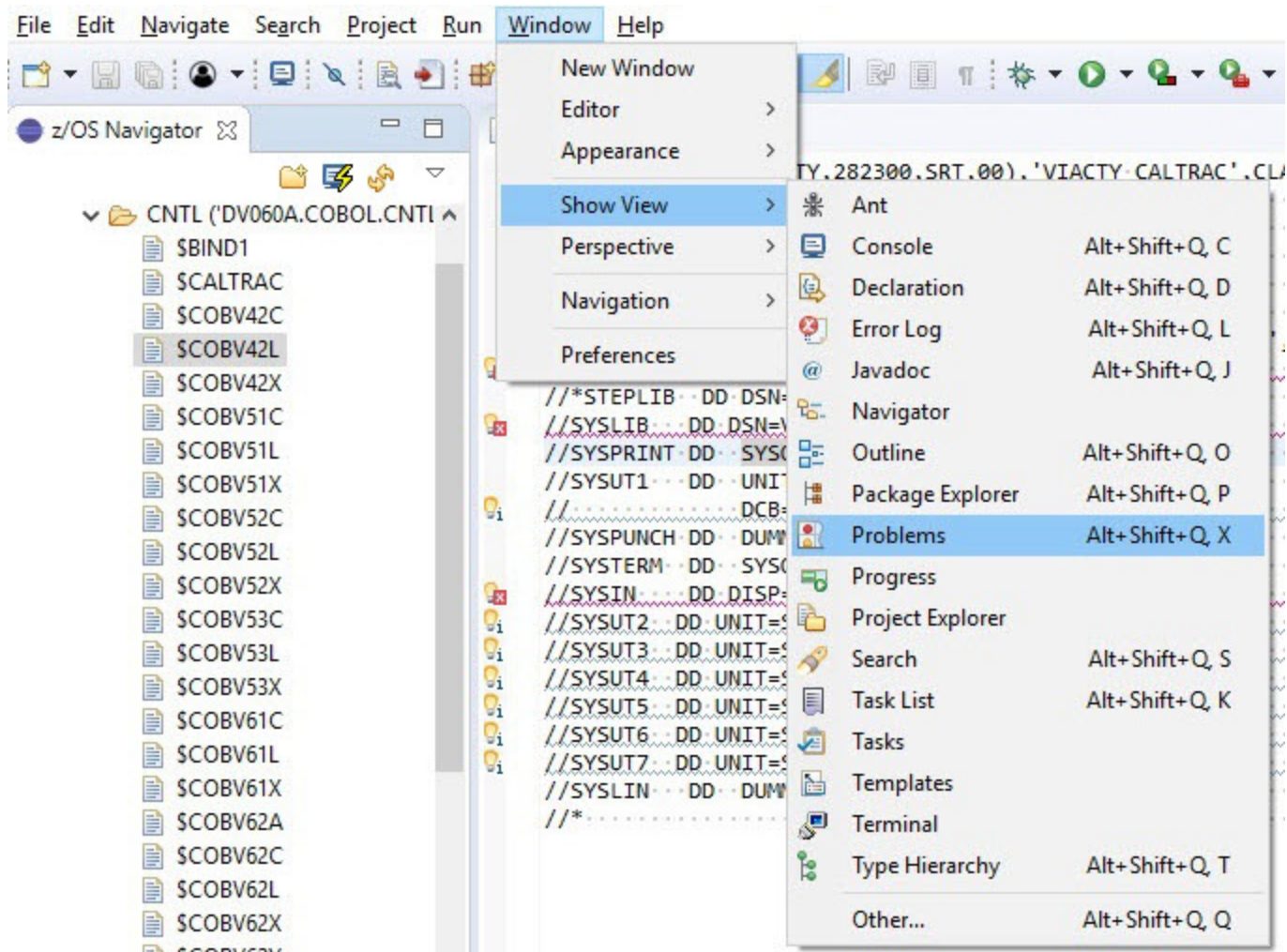
## Problems View

**Problems** view displays the list of problem messages that are identified in a JCL file. Generally, this view is automatically displayed when you scan a file. For more information about Problems view, refer to [Reports](#).

Description	Resource	Path	Location	Type
2 errors, 0 warnings, 401 others (Filter matched 102 of 403 items)				
Errors (2 items)				
DATA SET NOT FOUND IN CATALOG	SCALTRAC	/jclproject/DV060A....	line 10	JCL Scan Mes...
INVALID JOB PARAMETER - "DATE ".	SCALTRAC	/jclproject/DV060A....	line 2	JCL Scan Mes...
Infos (100 of 401 items)				
DSN = "VIASH71.PROD.COPYLIB """,{"error-id	SCALTRAC	/jclproject/DV060A....	line 10	JCL Scan Mes...
TEMPORARY DATA SET WILL BE USED.	SCOBV42C	/jclproject/DV060A....	line 65	JCL Scan Mes...
TEMPORARY DATA SET WILL BE USED.	SCOBV42C	/jclproject/DV060A....	line 65	JCL Scan Mes...
TEMPORARY DATA SET WILL BE USED.	SCOBV42C	/jclproject/DV060A....	line 65	JCL Scan Mes...
TEMPORARY DATA SET WILL BE USED.	SCOBV42C	/jclproject/DV060A....	line 65	JCL Scan Mes...

**To display Problems view:**

- From the menu, select **Window > Show View > Problems**.



## JCL Scan View

**JCL Scan** view displays the scanned report of a JCL file. This report is similar to the JOB/SCAN's Structured JCL Report (SJL). For more information about JCL Scan view, refer to [Reports](#).

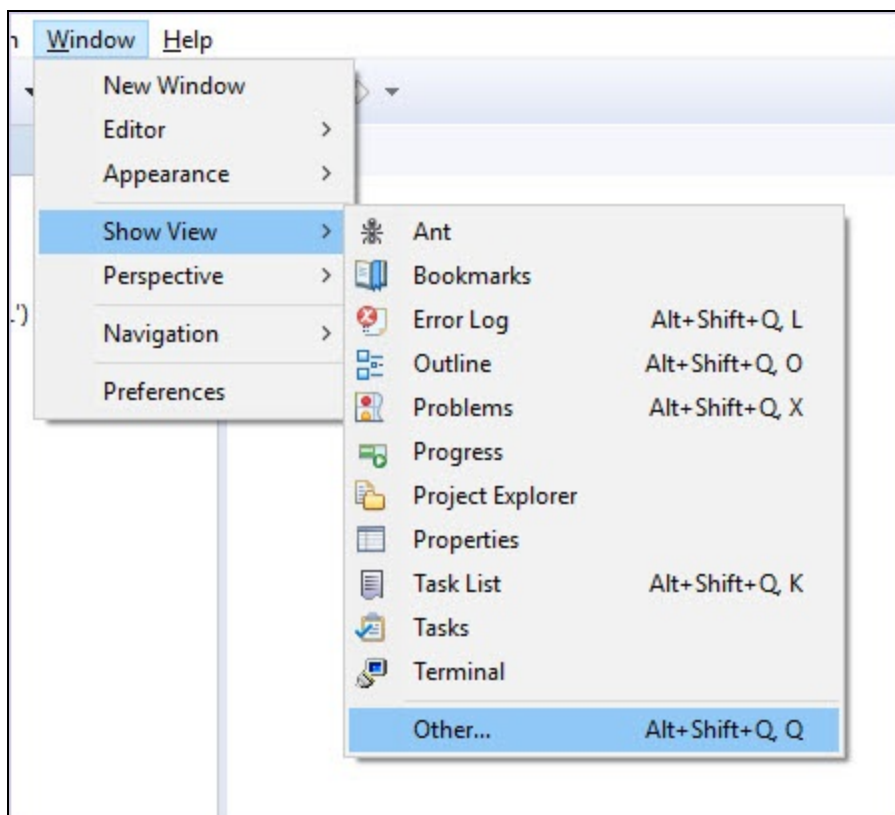
```

//CALTRAC JOB (DEVCTY,282300,SRT,00), 'VIACTY CALTRAC',CLASS=A,          00010000
// DATE=(,30),PRTY=6,MSGCLASS=X,NOTIFY=VIACTY                          00020000
***ERROR - DSS1080E - INVALID JOB PARAMETER - "DATE ".
//*                                                                       00030000
//* COMPILING VIASH71.PROD.COBOL(CALTRAC)                               00040000
//* USING VIASH71.PROD.COBOL($MACLIB$)                                  00050000
//CALTRAC EXEC PGM=IGYCRCTL,REGION=2M,                                  00060000
// PARM=('SIZE(MAX)', 'OPT', 'NORES', 'NODYN', 'NOADV', 'OBJ', 'APOST', 'LIB', 00070000
// 'MAP', 'NONUM', 'OFF', 'NOOBJ')                                     00080000
//*STEPLIB DD DSN=COB2.V402.COB2COMP,DISP=SHR                          00090000
//SYSLIB DD DSN=VIASH71.PROD.COPYLIB,DISP=SHR                          00100000
***ERROR - DSS4050E - DATA SET NOT FOUND IN CATALOG
*ADVISORY - DSS8900A - DSN = "VIASH71.PROD.COPYLIB
//SYSPRINT DD SYSOUT=*                                                00110000
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,2)),                              00120000
// DCB=(RECFM=FB.LRECL=80.BLKSIZE=32720)                              00130000

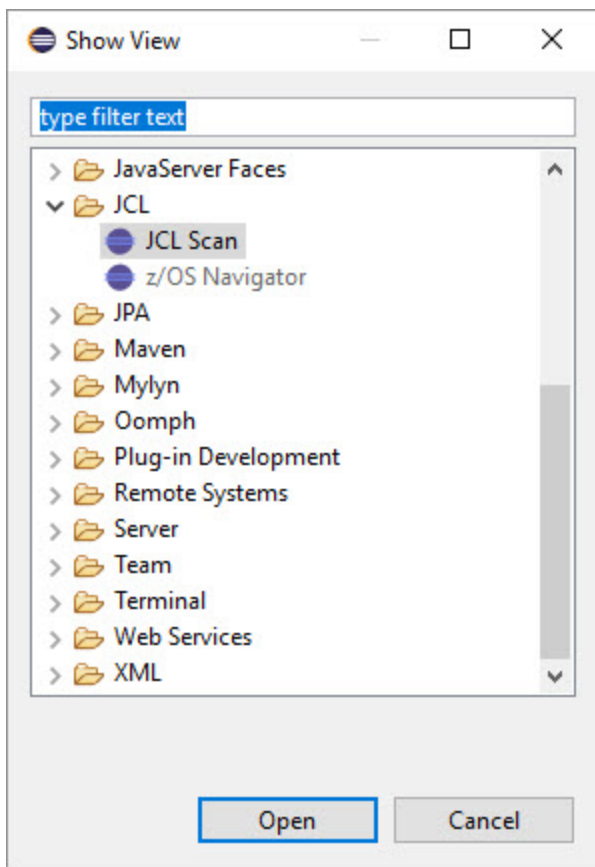
```

### To display JCL Scan view:

1. From the menu, select **Window > Show View > Other...**



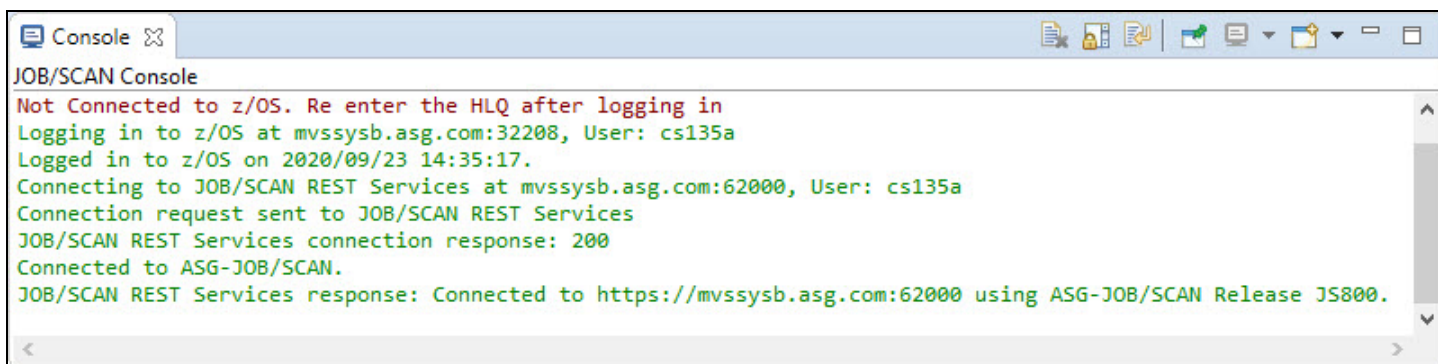
2. In the **Show View** dialog, do one of the following:
  - Enter **JCL Scan** in the **type filter text** box.
  - Locate and expand the **JCL** folder.



3. Double-click on **JCL Scan** or select **JCL Scan** and click **Open**.

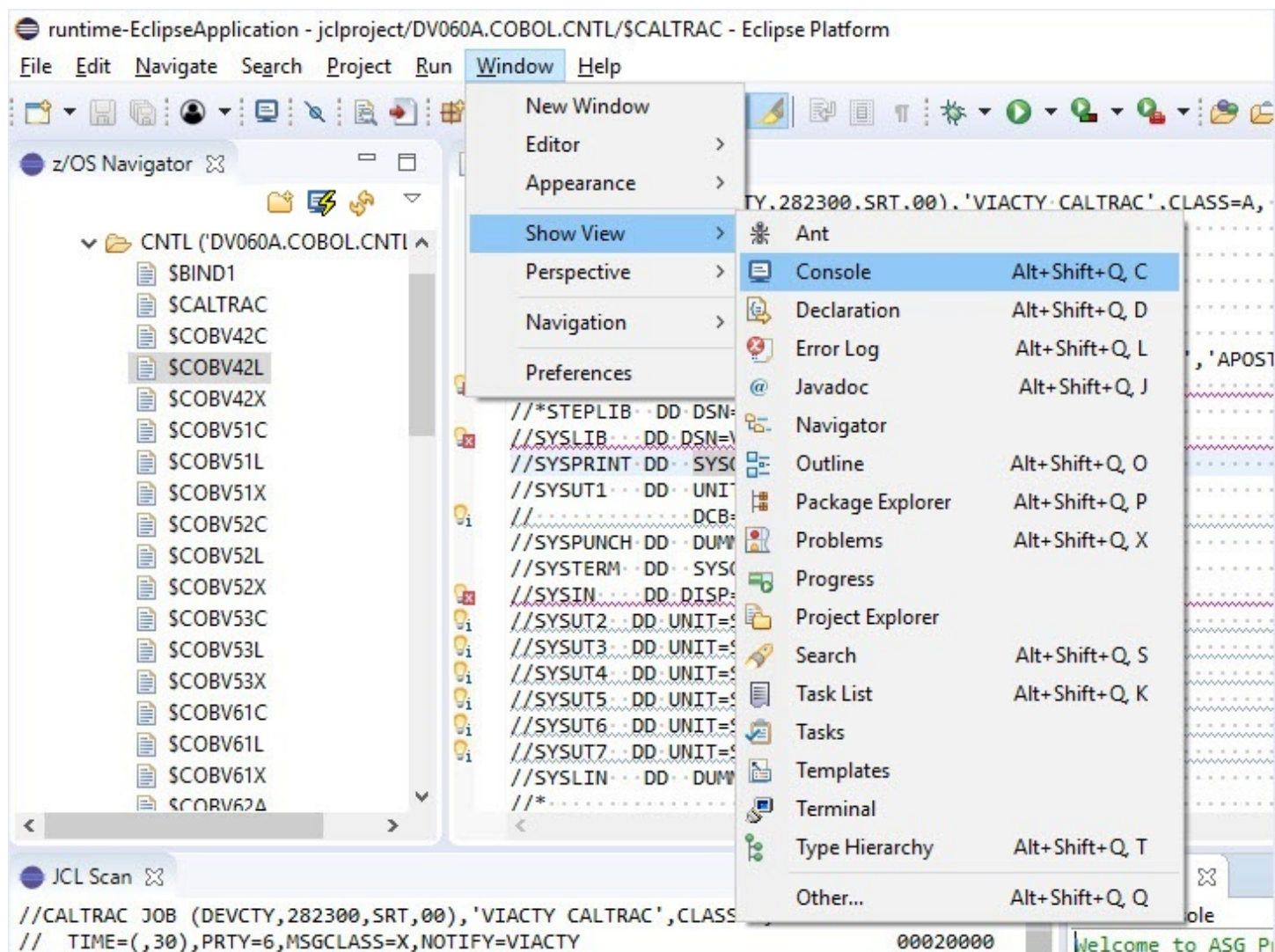
## JOB/SCAN Console View


**JOB/SCAN Console** view displays the output of a process performed in JOB/SCAN Plug-in. The console displays the output messages in two different colors: **Green** for the success messages and **red** for the error messages.



**To display JOB/SCAN Console view:**

1. From the menu, select **Window > Show View > Console**.



2. In **Console** dialog, click **Open Console** icon (  ) and select **JOB/SCAN Console**.





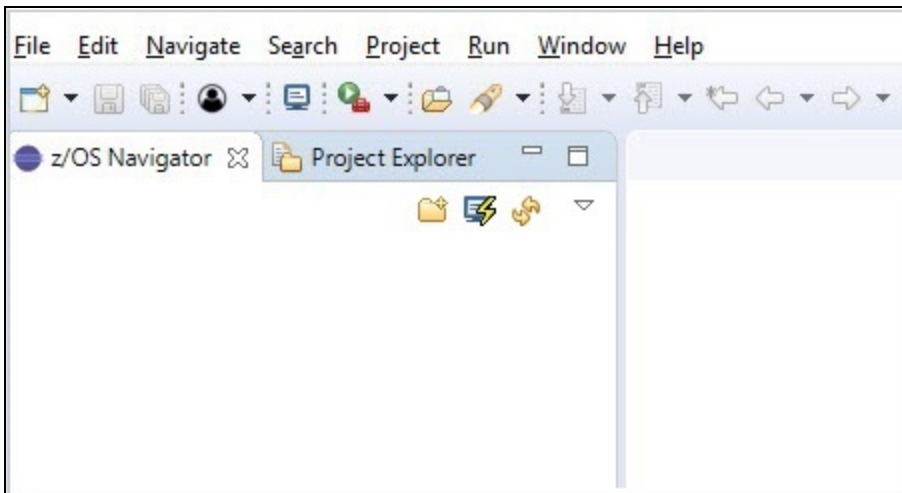
## Connecting to z/OS and ASG-JOB/SCAN

This topic explains the connection to z/OS system and ASG-JOB/SCAN. Connection to z/OS and JOB/SCAN is essential to view z/OS resources and perform JOB/SCAN functionalities.

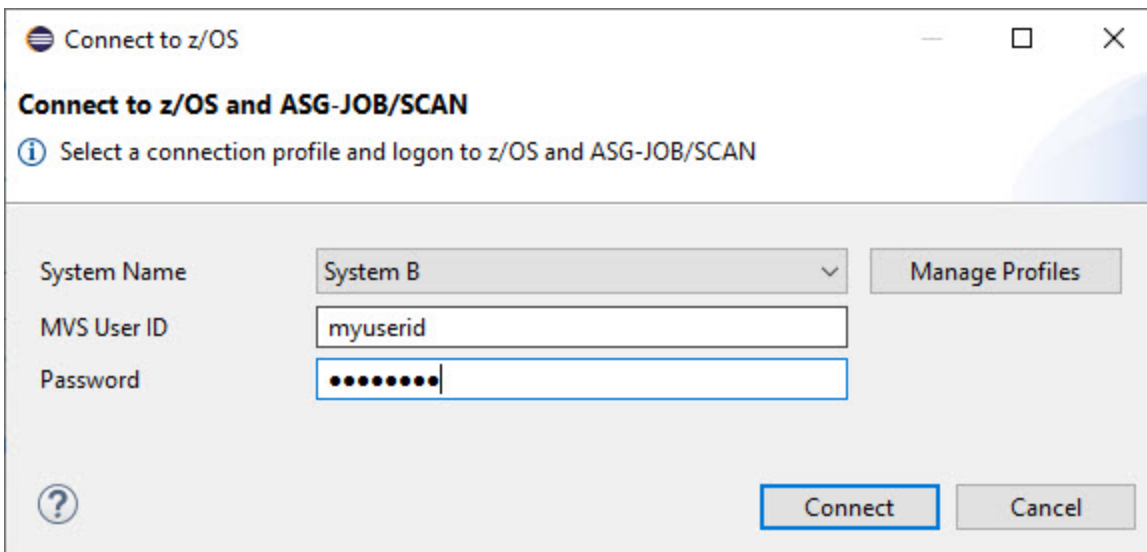
If you are a first time user, then you must create a connection profile that includes information about the z/OS system and ASG-JOB/SCAN.

### To connect to z/OS:

1. Select  (**Connect to z/OS**) icon from the **z/OS Navigator** pane.



The **Connect to z/OS** dialog appears.



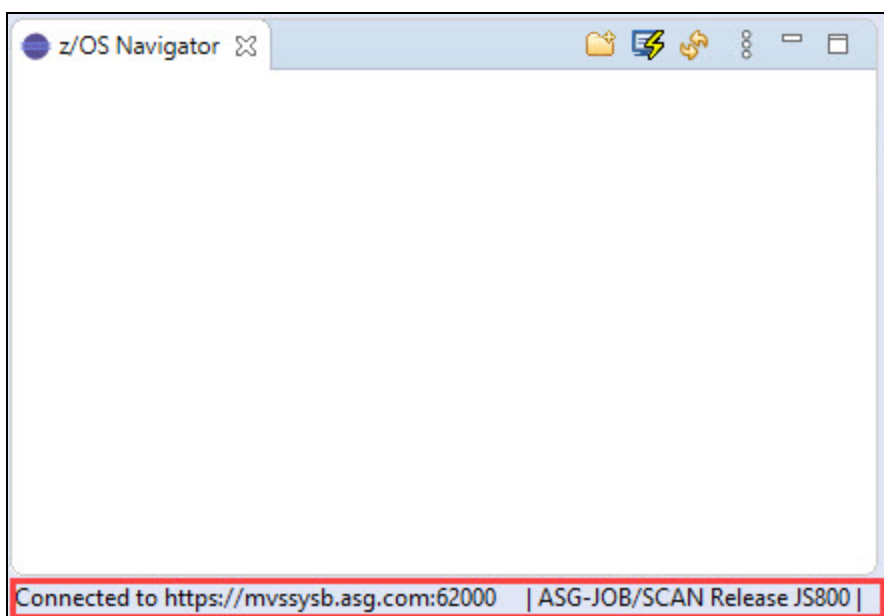
2. Select **System Name** from the list of connection profiles. **MVS User ID** is auto-populated on selecting the system name.




Initially, the **System Name** is empty. Create a new connection profile to display your profile in the **System Name** list. See the sub-topic: [To create a connection Profile](#) for more information.

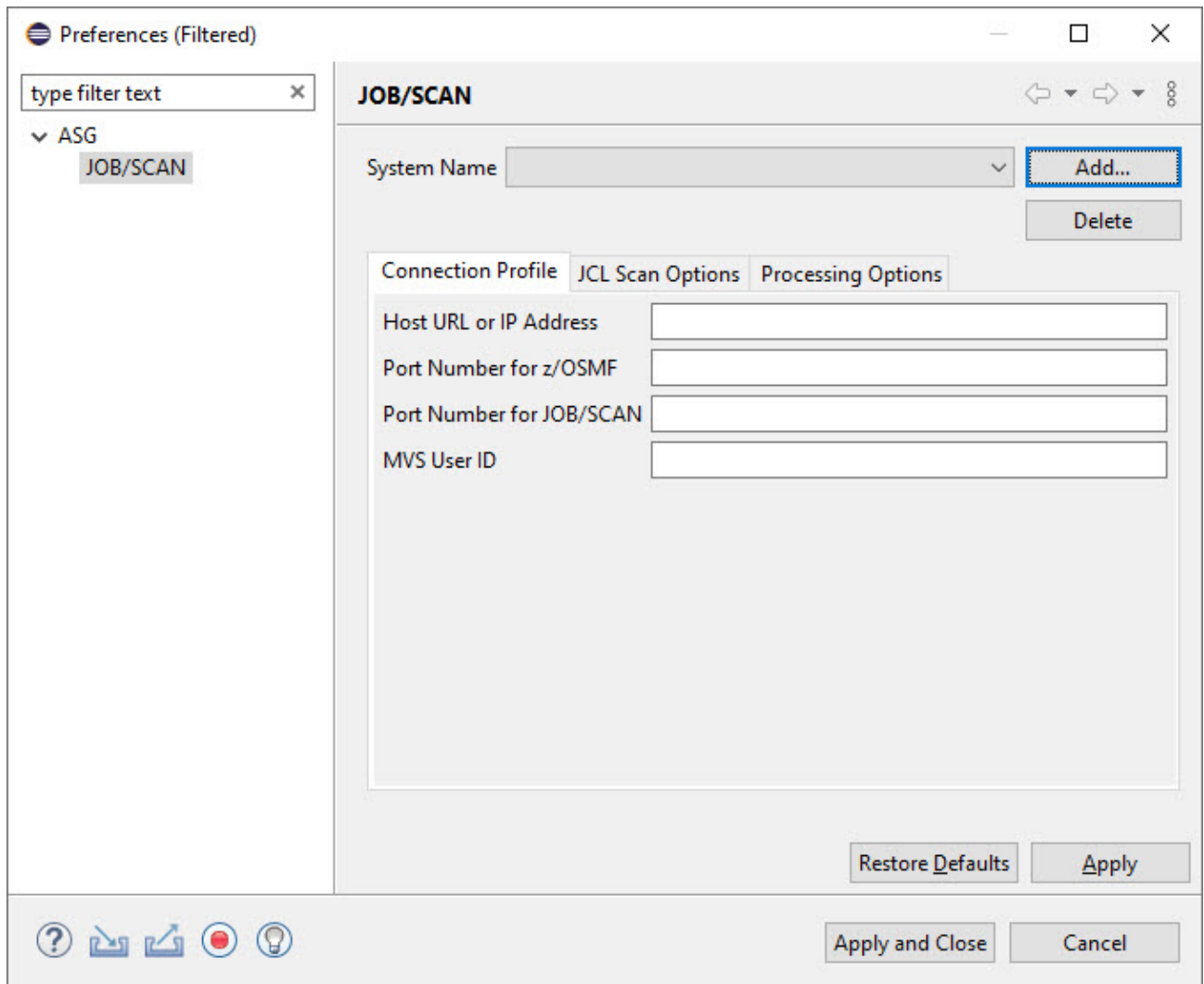
3. Enter **Password** and click **Connect**.

If logon is successful, the **Connect to z/OS** dialog is closed and the connection status along with the version of the JOB/SCAN is displayed in the status bar. Else, a self-explanatory error message appears in the **Connect to z/OS** dialog.

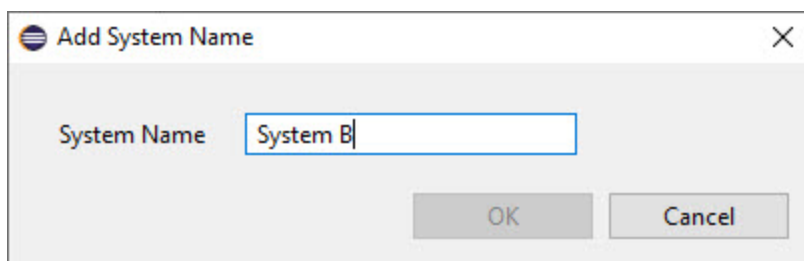


#### **To create a Connection Profile:**

1. Do either of the following to navigate to the **JOB/SCAN** preference page:
  - From the **z/OS Navigator** pane, select  (Connect to z/OS) icon. In the **Connect to z/OS** dialog, click **Manage Profiles**.
  - From the **Window** menu, select **Preferences**. In the **Preferences** dialog, select **ASG > JOB/SCAN**.
2. Do the following to create a new system name:
  - i. Click **Add....**The **Add System Name** dialog appears.



- ii. Enter a name for the new connection profile and click **OK**.



The given system name appears in the **JOB/SCAN** preference page.

3. In the **Connection Profile** tab fill the following details:

Preferences (Filtered)

type filter text

ASG

JOB/SCAN

**JOB/SCAN**

System Name: System B

Add...  
Delete

Connection Profile | JCL Scan Options | Processing Options

Host URL or IP Address: mvssysb.asg.com

Port Number for z/OSMF: 32208

Port Number for JOB/SCAN: 62000

MVS User ID: myuserid

Restore Defaults | Apply

Apply and Close | Cancel

Field Name	Description
Host Name or IP Address	Enter host address or IP address of the z/OS system.
Port Number for z/OSMF	Enter port number of the host machine on which the z/OSMF REST API service is running.
Port Number for JOB/SCAN	Enter port number of the host machine on which the JOB/SCAN REST API service is running.
MVS User ID	Enter User ID for logging on to the z/OS system.

4. Click **Apply** to save the scan options or click **Apply and Close** to save the scan options and return to the **Connect to z/OS** dialog.

**To update a Connection Profile:**

1. In the **JOB/SCAN** preference page, select the **System Name** that you like to edit and update the details.
2. Click **Apply** to save the connection profile or click **Apply and Close** to save the connection profile and return to the **Connect to z/OS** dialog.



After updating the connection profile, save the changes and reconnect to ASG-JOB/SCAN using the **Connect to z/OS** dialog to establish connection with updated details.

**To delete a Connection Profile:**

1. In the **JOB/SCAN** preference page, select the **System Name** that you like to delete and click **Delete**.
2. In the confirmation dialog, click **OK**. Your connection profile is deleted from the **System Name** list.



The connection profile details and scan options that are saved in the System Name will be lost, so make sure before you delete it.

# Setting the JCL Scan Options

This topic explains how to set scan options that are used when validating JCLs. Similar to ASG-JOB/SCAN, the ASG-JOB/SCAN Eclipse Plug-in allows you to specify some options that are used when validating the JCLs. You can set the following options in ASG-JOB/SCAN Plug-in:

## First PROCLIB

When validating JCL, it might be necessary to specify a PROC library to be searched before any other libraries. The Scan Options tab provides a First PROCLIB field for this purpose. When this field is set, JOB/SCAN will look for the PROC and INCLUDE members used in the input JCL file in the library first before searching other libraries. The only type of library supported in this release is PDS.

## Selection Exit

A Selection Exit is a user-defined program that processes the input JCL before JOB/SCAN parses and validates it. The exit evaluates the JCL, makes changes if necessary, and issues a return code instructing JOB/SCAN on the action to be taken with the JCL. The exits can be written in any LE-compliant language. For sample selection exit programs, refer to CA7J1 and CA7J2 members in the SJJSSAMP library. For more information on Selection Exit, see [Selection Exit Interface](#).

## Standards

In addition to the general error validation, the JOB/SCAN's Standards Facility provides the ability to integrate site's additional requirements for validating JCLs. Standards (standards exits or standards routines) are a set of rules that enforces site-specific requirements into JOB/SCAN's JCL validation. The standards are user-defined programs that can be written in any language including OS/VS COBOL, VS COBOL II, REXX, and Assembler. Sample programs are available in the SJJSTDL library. For a detailed introduction to JOB/SCAN's standards facility, see [Standards Information](#).

### *To set the JCL scan options:*

1. Do either of the following to navigate to the **JOB/SCAN** preference page:
  - From the **z/OS Navigator** pane, select (Connect to z/OS) icon. In the **Connect to z/OS** dialog, click **Manage Profiles**.
  - From the **Window** menu, select **Preferences**. In the **Preferences** dialog, select **ASG > JOB/SCAN**.
2. Make sure that the correct **System Name** is selected or add a new one by clicking **Add....**

Preferences (Filtered)

type filter text

ASG

JOB/SCAN

**JOB/SCAN**

System Name System B Add... Delete

Connection Profile **JCL Scan Options** Processing Options

Host URL or IP Address mvssysb.asg.com

Port Number for z/OSMF 32208

Port Number for JOB/SCAN 62000

MVS User ID myuserid

Restore Defaults Apply

Apply and Close Cancel

3. Make sure that the **Connection Profile** fields are filled in. See [Connecting to z/OS and ASG-JOB/SCAN](#) for more information.
4. Select the **JCL Scan Options** tab and fill the following details as per your requirement:

The screenshot shows the 'Preferences (Filtered)' window for 'JOB/SCAN'. The 'System Name' is 'System B'. Under the 'JCL Scan Options' tab, the 'Include summary in JCL scan output' checkbox is checked. The 'First PROCLIB' is set to 'ASGJJS.TRY01.PROCLIB'. The 'Selection Exit' is 'SEL01' and its 'Parameters' are 'PARM24'. The 'Standards Name' is 'STD01' and its 'Parameters' are 'PARM12'. Buttons for 'Add...', 'Delete', 'Restore Defaults', 'Apply', 'Apply and Close', and 'Cancel' are present.

Field Name	Description
Include summary in JCL scan output	Select to display the JCL scan options in the JCL scan report.
First PROCLIB	Enter the library that are searched first before other libraries for PROC and INCLUDE members.
<b>Selection Exit</b>	
Selection Exit	Enter the name of a selection exit program.
Parameter	Enter any parameters that is required for the execution of the selection exit program.
<b>Standards</b>	



Field Name	Description
Standards Name	Enter the name of a standards program.
Parameter	Enter any parameters that is required for the execution of the standards program.

5. Click **Apply** to save the scan options or click **Apply and Close** to save the scan options and return to the **Connect to z/OS** dialog.

# Setting the Processing Options

This topic explains the procedure to set the processing options that control how JOB/SCAN processes the input JCL, customize various reports and third-party interfaces. These options are defined in the Preferences page. In ASG-JOB/SCAN, user can enter the values for these options on the Processing Options panel or can code in the SJJSPARM library. Values entered on the panel override corresponding parameters coded in the SJJSPARM library. Similarly, values entered in the Plug-in's Preference page will override corresponding parameters coded in JOB/SCAN's SJJSPARM library. You can set the following processing options:

## Use Error Table

JOB/SCAN provides an option to suppress or change the severity level of JCL error messages for a specified JOB/SCAN error numbers. The entries in the Control Table AW are used to suppress any error messages or change their severity level. When the Use Error Table option is selected, JOB/SCAN searches the Control Table AW to determine if any error messages issued on the JCL statement are listed. If the error number is found in Control Table AW, then the severity is changed to the level indicated or is optionally suppressed.

## Condition Code Processing

JOB/SCAN provides an option to control the effect of CONDCODE parameters during execution. The condition code validation is done for COND= parameters on the job statement, EXEC PROC statement, EXEC PGM statements, and for // IF statements. Control Table AU contains Job steps to be ignored by JOB/SCAN. Control Table AR contains the return codes. When the Condition Code Processing option is selected, then JOB/SCAN excludes all program steps identified in Control Table AU entries, and each step is assigned a return code 0 unless a Control Table AR entry is specified.

## Catalog Access

The Catalog Access option controls the access to the system catalog. When this is selected, JOB/SCAN is allowed to access the system catalog.

## VTOC Access

The VTOC Access option controls the access to the DASD VTOCs. When this is selected, JOB/SCAN is allowed to access the DASD VTOCs.

## PDS Member Verification

The PDS Member Verification option determines whether or not JOB/SCAN can validate partitioned dataset member names. Verification is done when a DSN=dsname(membername) parameter is coded in the JCL DD statement.

## Security Exit DSNAME Check

The Security Exit DSNAME Check option allows JOB/SCAN to detect access authority violations. When this option is selected, JOB/SCAN instructs the security exit program to verify that the user ID running JCL validation has the necessary access authority to the datasets that are mentioned in the JCL before actually attempting to access the file.



When this option is selected and if there is no security exit active or no DSNNAME-checking support in the exit, then JOB/SCAN ignores this option.

## Program Checking

The Program Checking option determines whether or not JOB/SCAN can validate the existence of programs. If this is selected, JOB/SCAN is allowed to validate the programs.

## Disable SMS Processing

The Disable SMS Processing option determines whether or not JOB/SCAN can access the Automatic Class Selection (ACS) routines and the Construct Access Services (CAS) of the MVS DFSMS subsystem. If this option is selected, JOB/SCAN will bypass all SMS processing. If deselected, the SMS Run option specified in the SJJSPARM library will be in effect.

## IDCAMS Processing

JOB/SCAN has the ability to validate IDCAMS control statements when found within a JCL member. If this option is selected, JOB/SCAN validates IDCAMS control statements and tracks datasets created or deleted by the IDCAMS.

## Use SCHEDULER Interface

When jobs are executed using a scheduler, it allows the JCL to contain environmental variables which must be resolved at the time of job submission. JOB/SCAN supports a wide range of schedulers and it also resolves the environmental variables before the job is submitted. Select this option to turn on the scheduler interface specified for the SCHEDULER Run option in the SJJSPARM library.

## Access Control Cards

A Control Card is a dataset or an inline statement(s) that control how the validation should be processed. When this option is selected, JOB/SCAN is directed to read members within the job stream that contain control cards, verify control card input, and include control cards in the JCL Scan report. You can also limit the number of control cards listed in the JCL Scan view for each DDNAME or dataset name identified as control card input.

## Subsystem

The input JCL may include statements related to Job Entry Subsystem(JES) that need to be processed by JOB/SCAN. This option indicates whether or not JOB/SCAN can validate the JES statements. If JES2 or JES3 is selected, then JOB/SCAN searches for any non-IBM-standard JES statements and uses the entries in Control Table AS to standardize the JES statements.

## Suppress Error Messages

JOB/SCAN provides an option to suppress error messages that are normally displayed in the JCL scan reports. This is done based on the message type. The JOB/SCAN plug-in provides the following options to suppress the messages from being displayed in the JCL Scan view, the Problems view, and the JCL editor:

- **do not suppress:** display all messages
- **advisory only:** suppress advisory messages
- **advisory and warning:** suppress warning and advisory messages
- **advisory, warning, and error:** suppress advisory, warning, and error messages
- **less than or equal to:** suppress all error messages with a severity level equal to or less than the specified value

## Extended Error Text

In JOB/SCAN, you can display a messages in two ways: short message text and extended message text. The short error text is a single-line messages that provide less information about the error. The extended messages text give users a better understanding of the violation beyond what the basic short error text provide. This option allows you to select the type of message text to be displayed for each error in the result of a scan.



For more information on processing options, refer to [ASG-JOB/SCAN Reference Information](#).

### *To set the JCL scan options:*

1. Do either of the following to navigate to the **JOB/SCAN** preference page:
  - From the **z/OS Navigator** pane, select (Connect to z/OS) icon. In the **Connect to z/OS** dialog, click **Manage Profiles**.
  - From the **Window** menu, select **Preferences**. In the **Preferences** dialog, select **ASG > JOB/SCAN**.
2. Make sure that the correct **System Name** is selected or add a new one by clicking **Add...**

Preferences (Filtered)

type filter text

ASG

JOB/SCAN

**JOB/SCAN**

System Name System B Add... Delete

Connection Profile JCL Scan Options **Processing Options**

Host URL or IP Address mvssysb.asg.com

Port Number for z/OSMF 32208

Port Number for JOB/SCAN 62000




MVS User ID myuserid

Restore Defaults Apply

Apply and Close Cancel

3. Make sure that the **Connection Profile** fields are filled in. See [Connecting to z/OS and ASG-JOB/SCAN](#) for more information.
4. Select the **Processing Options** tab and fill the following details as per your requirement:

Field Name	Description
Use Error Table	Select to access Control Table AW for error suppression based on severity levels.
Condition Cod Processing	Select to request that JOB/SCAN evaluate all the CONDCODE parameters.
Catalog Access	Deselect to inhibit access to the system catalog.
VTOC Access	Deselect to inhibit access to the DASD VTOCs.
PDS Member Verification	Select to verify the existence of PDS members referenced in the JCL.
	<div style="border: 1px solid purple; padding: 5px; display: inline-block;">  This option is disabled if <b>VTOC Access</b> is not selected.         </div>

Field Name	Description
Security Exit DSNNAME Check	Select to invoke the security exit to check each dataset name in the input JCL for possible security violations.
Program Checking	Select to validate the existence of programs.
Disable SMS Processing	Select to bypass all SMS processing.  This option has no effect if the SMS subsystem is not active.
IDCAMS Processing	Select to validate IDCAMS control statements.
Use SCHEDULER Interface	Select to turn on the SCHEDULER Run Option in JOB/SCAN's SJJSPARM library.
Access Control Cards	Select to access control card libraries, verify control card input, and include control cards in the validation.  This option is enabled only if both the <b>Catalog Access</b> and <b>VTOC Access</b> options are selected.
Control Card Limit	Limit the number of control cards (1-999) listed in the JCL Scan view for each DDNAME or dataset name identified as control card input.  This field is enabled only if <b>Access Control Cards</b> is enabled and selected.
Job Limit	Limit the number of jobs to be processed (1-999) in each JCL scan.
Subsystem	Use <b>JES2</b> , <b>JES3</b> , or <b>none</b> to indicate whether the JCL contains JES2 or JES3 statements to be validated for syntax. JES statements are not evaluated if <b>none</b> is selected for this option.
Suppress Error Messages	Select the type of messages to suppress in the result of a scan, including the JCL Scan view, the Problems view, and the JCL editor. The options are: <ul style="list-style-type: none"> <li>do not suppress</li> <li>advisory only</li> <li>advisory and warning</li> <li>advisory, warning, and error</li> <li>less than or equal to</li> </ul>
Severity Level	Select the severity level below or at which error messages are suppressed. This selec-

Field Name	Description
Extended Error Text	<p>tion is enabled only if <b>less than or equal to</b> is selected for <b>Suppress Error Messages</b> option.</p> <p>Select the type of message text to be displayed for each error in the result of a scan. The options are:</p> <ul style="list-style-type: none"><li>• short message text</li><li>• extended message text</li></ul>

- 
5. Click **Apply** to save the scan options or click **Apply and Close** to save the scan options and return to the **Connect to z/OS** dialog.



# Working with z/OS Datasets


This topic explains how to:

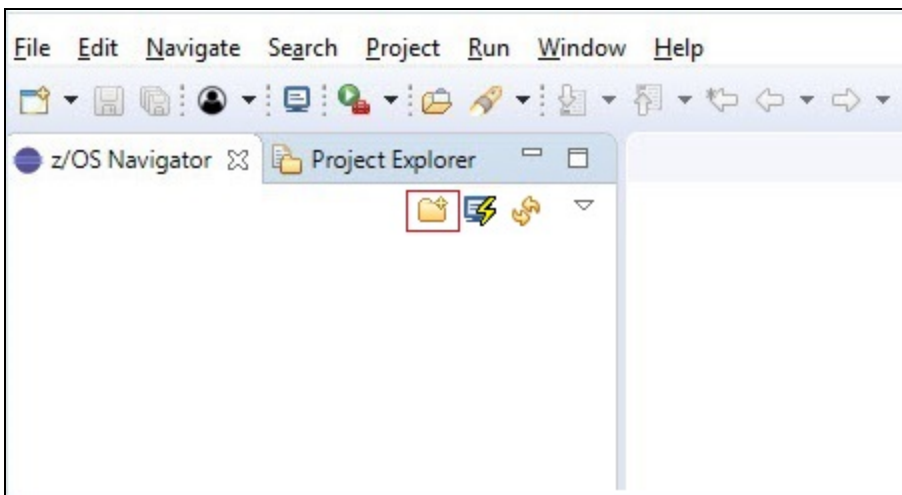
- display the z/OS datasets
- edit the z/OS datasets
- refresh the list of datasets

## Displaying Datasets

Datasets that are in the z/OS system can be explored from the z/OS Navigator view.

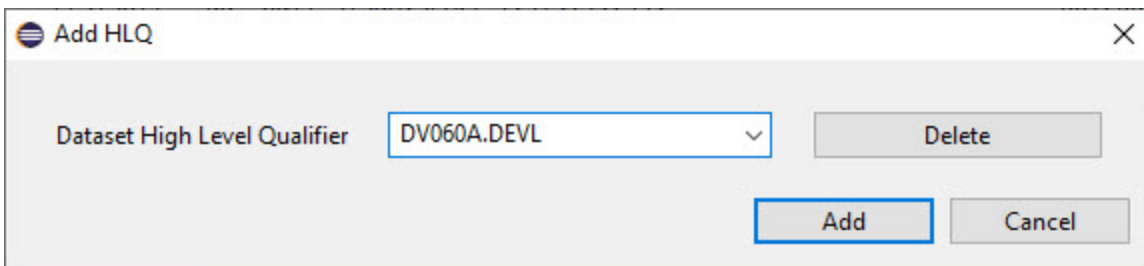
### To display datasets:

1. Select  (Add High Level Qualifier) icon from the **z/OS Navigator** pane. The **ADD HLQ** dialog box appears.



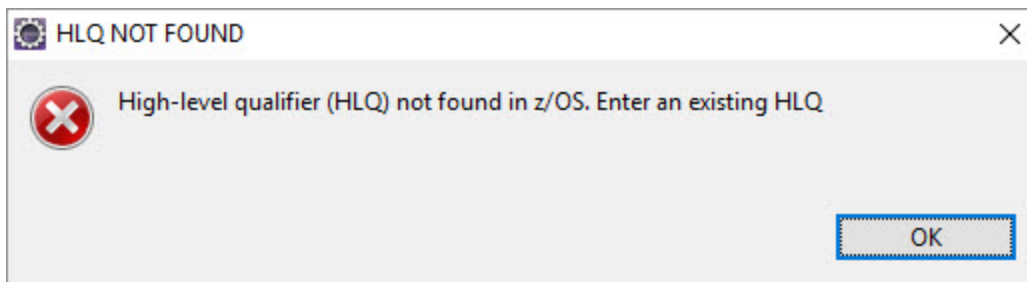
If connection to z/OS is not established, then **Connect to z/OS** dialog appears and prompts for main-frame credentials. See [Connecting to z/OS and ASG-JOB/SCAN](#) for more information.

2. In the **Dataset High Level Qualifier**, enter/ select the dataset's high-level qualifier that you would like to view and click **Add**.

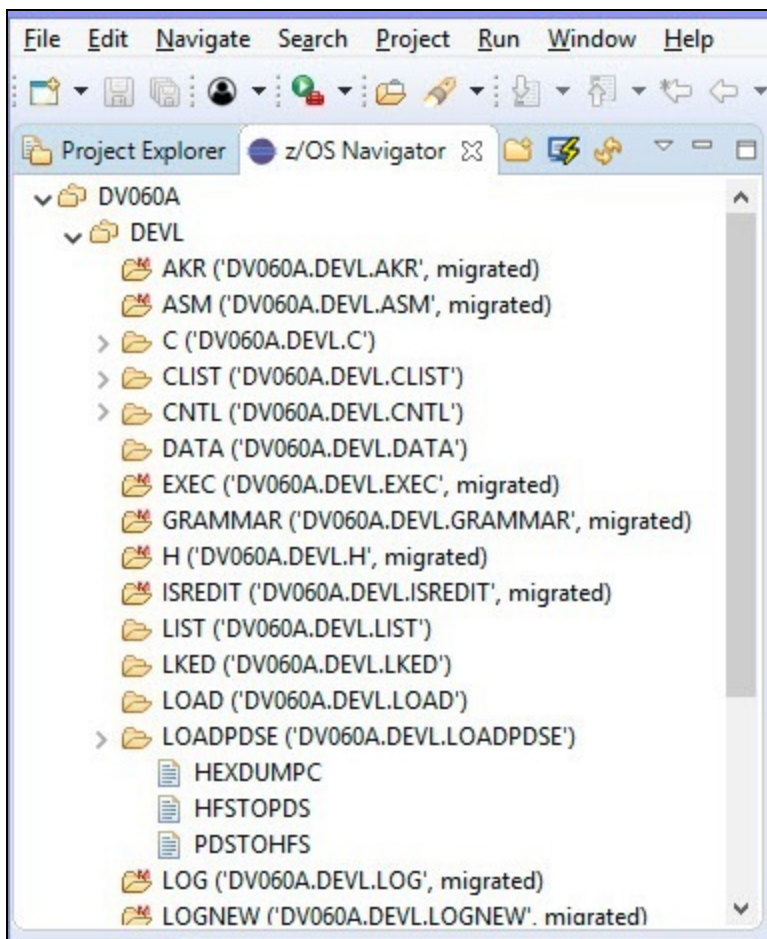




- If you have entered the high-level qualifier for the first time, then on clicking **Add**, the HLQ is saved to the **Dataset High Level Qualifier** list. The list keeps track of the HLQs that you have entered in the past so you don't have to retype again.
- To delete a saved HLQ from the **Dataset High Level Qualifier** list, select the HLQ and click **Delete**.
- If the entered HLQ is not found in z/OS, then the following error message is displayed. Make sure to enter an existing HLQ.



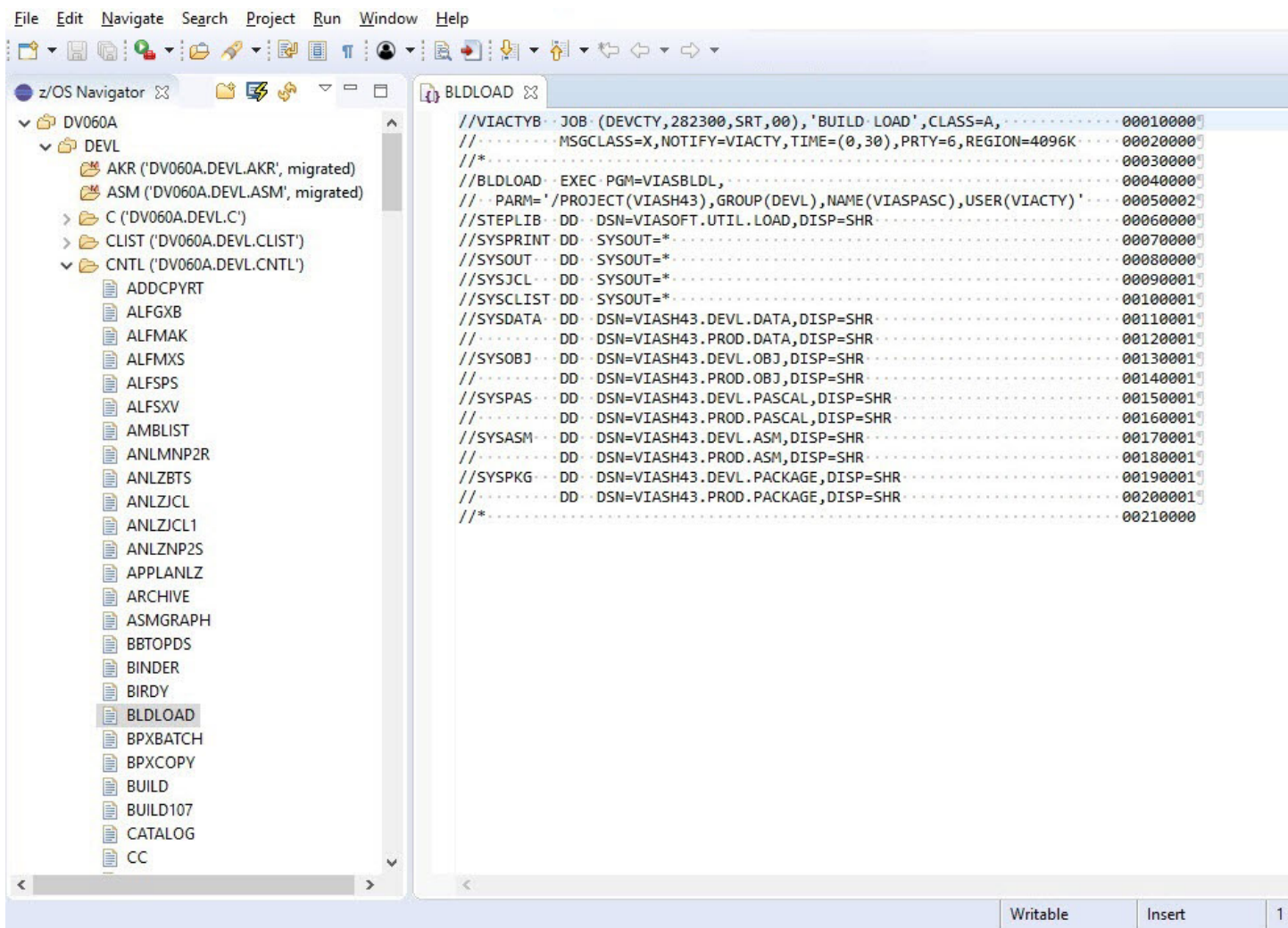
3. Datasets are displayed under a High-Level Qualifier (HLQ) in a tree-like structure. Expand the HLQ to view the files.



Icons in the tree structure represent the following:

Icon	Description
	node in a dataset qualifier
	PDS or PDSE containing dataset members
	sequential dataset
	partitioned dataset that has been migrated
	sequential dataset that has been migrated
	member in PDS or PDSE

- Double-click on a dataset, the z/OS navigator opens the file in the JCL Editor. A tab in the JCL editor displays the name of the file, while its contents appear in the main region of the tabbed pane.

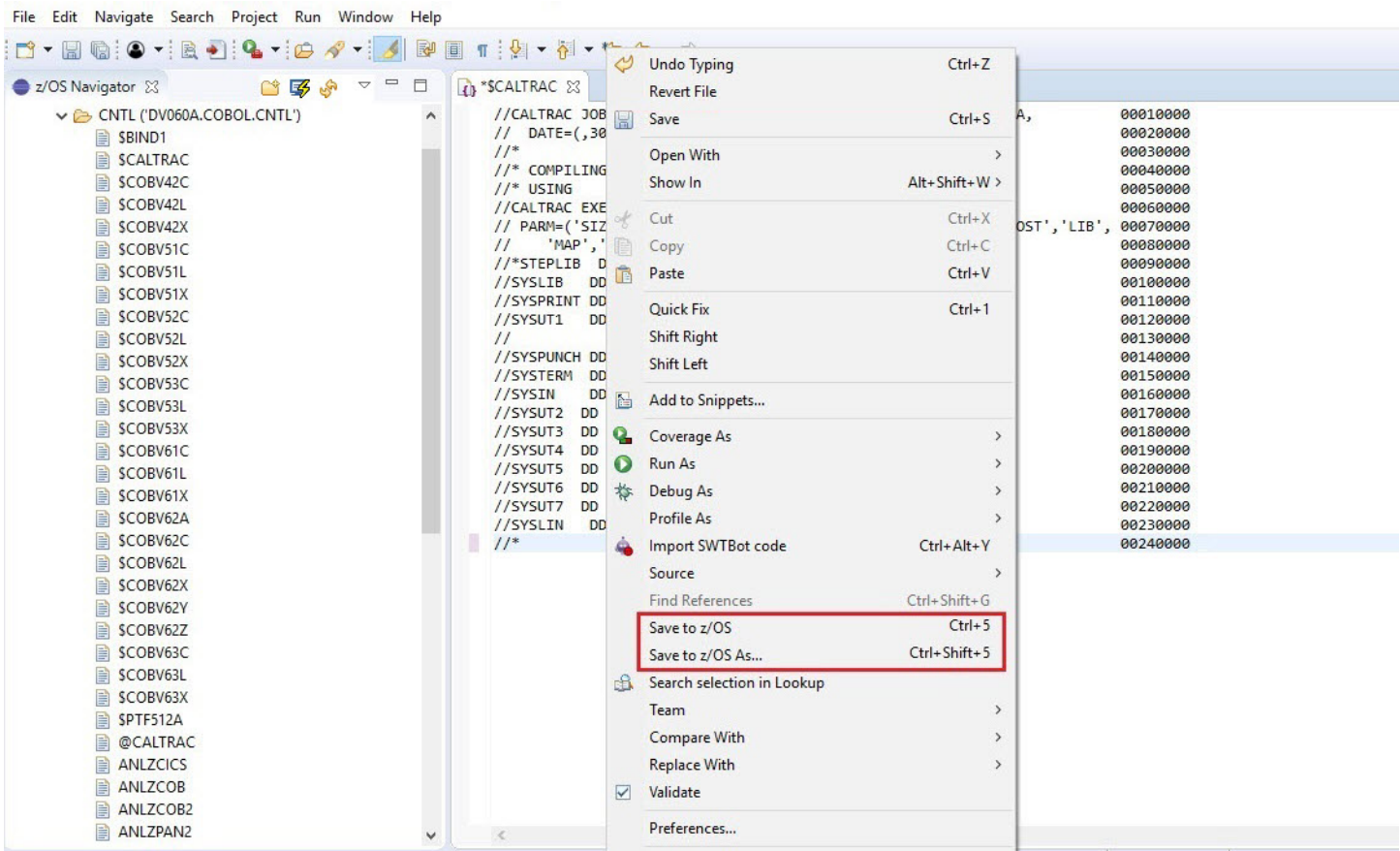


## Editing and Saving Datasets

The JCL Editor allows the user to edit and save a dataset.

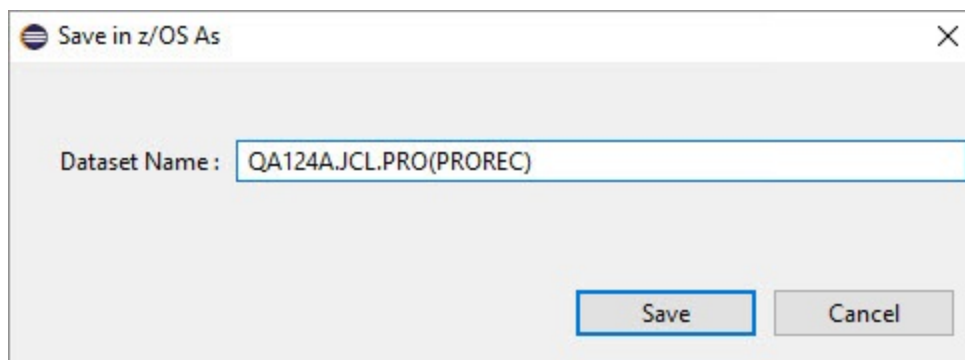
### **To edit and save a dataset:**

1. In the **z/OS Navigator** view, double-click on a dataset that you like to update. The dataset is displayed in the JCL Editor.
2. After updating the dataset, right-click in the JCL editor area to bring up the dataset contextual menu.



3. Select one of the options to save the changes to z/OS:

- a. **Save to z/OS** - This option allows the user to save the file under the same dataset name.
- b. **Save to z/OS As...** - This option allows the user to save the file either as a new member in a PDS or as an existing PS file. When this option is selected, user is prompted to enter a dataset name. Enter a name and click **OK**.



★ This option will not allocate a new dataset.

4. Refresh the z/OS Navigator to view the changes.

## Refreshing Datasets

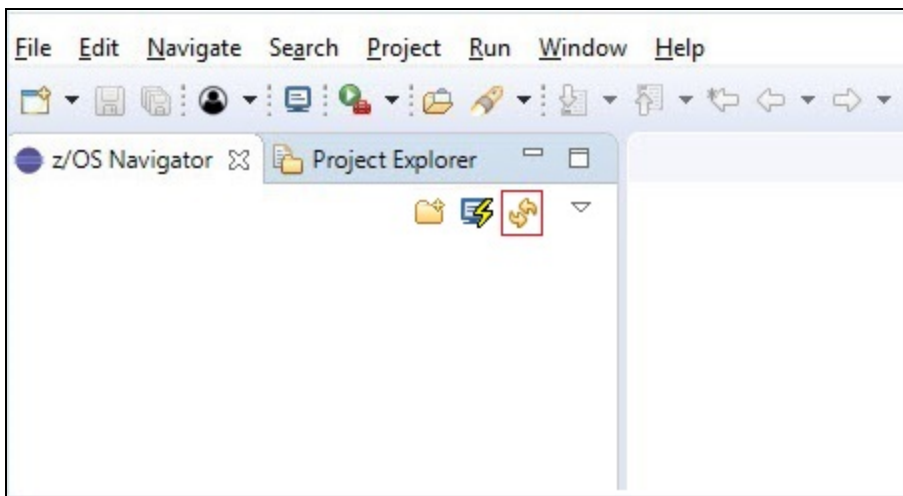
The z/OS Navigator view provides a refresh icon that refreshes the list of datasets.

You may need to refresh dataset list whenever you:

- save a file to z/OS
- add / delete / edit a dataset or a member in z/OS

### **To refresh datasets:**

- Click the 🔄 (Refresh View) icon from the **z/OS Navigator** pane.



See [Validating JCL](#) for information on scanning / validating a file.

# Working with Local Files

ASG-JOB/SCAN Plug-in allows the user to scan files that are present in your local file system. This topic explains how to work with the files that are in your local file system. The files in your local system can be accessed from the Project Explorer view.

## Displaying Local Files

To display local files from the Project Explorer view, first they must be imported. You can import the files by one of the following methods:

- dragging and dropping from the file system
- copying and pasting from the file system
- using the Import option

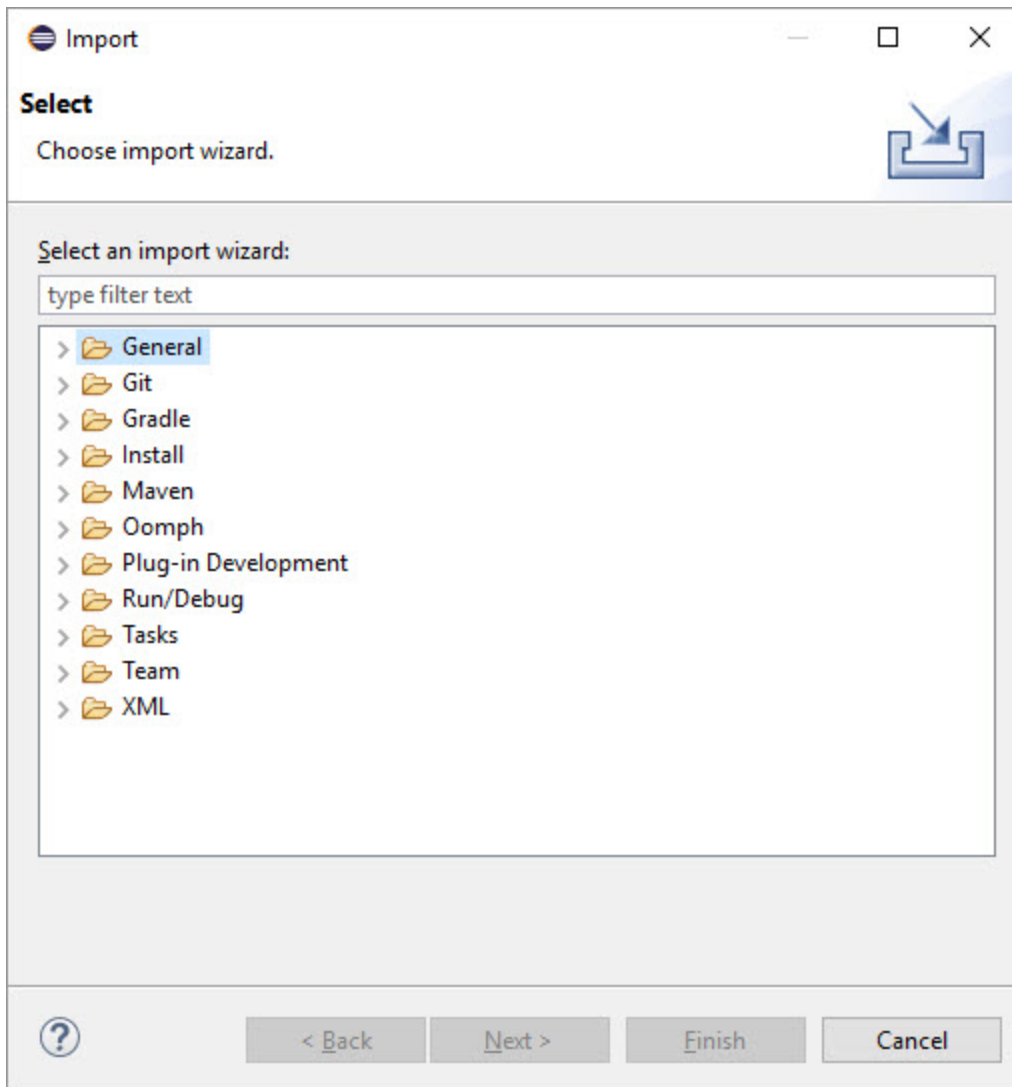
After importing the files, right-click on the file that you like to view and select **Open With --> JCL Editor**.



- If you don't find the **JCL Editor** option, select **Other...** and in the **Editor Selection** dialog, select **JCL Editor** from the **Internal Editors** list.
- When a file is opened from **Project Explorer** view, JOB/SCAN commands are available in JCL Editor **only**. The commands are displayed as buttons in the toolbar as well as additional options in the **Edit** menu. If you are not already connected to a JOB/SCAN server when you run the JOB/SCAN commands, you will be asked to connect to JOB/SCAN first before the commands are executed. Perform the steps in [Connecting to z/OS and ASG-JOB/SCAN](#) to connect to JOB/SCAN's server.

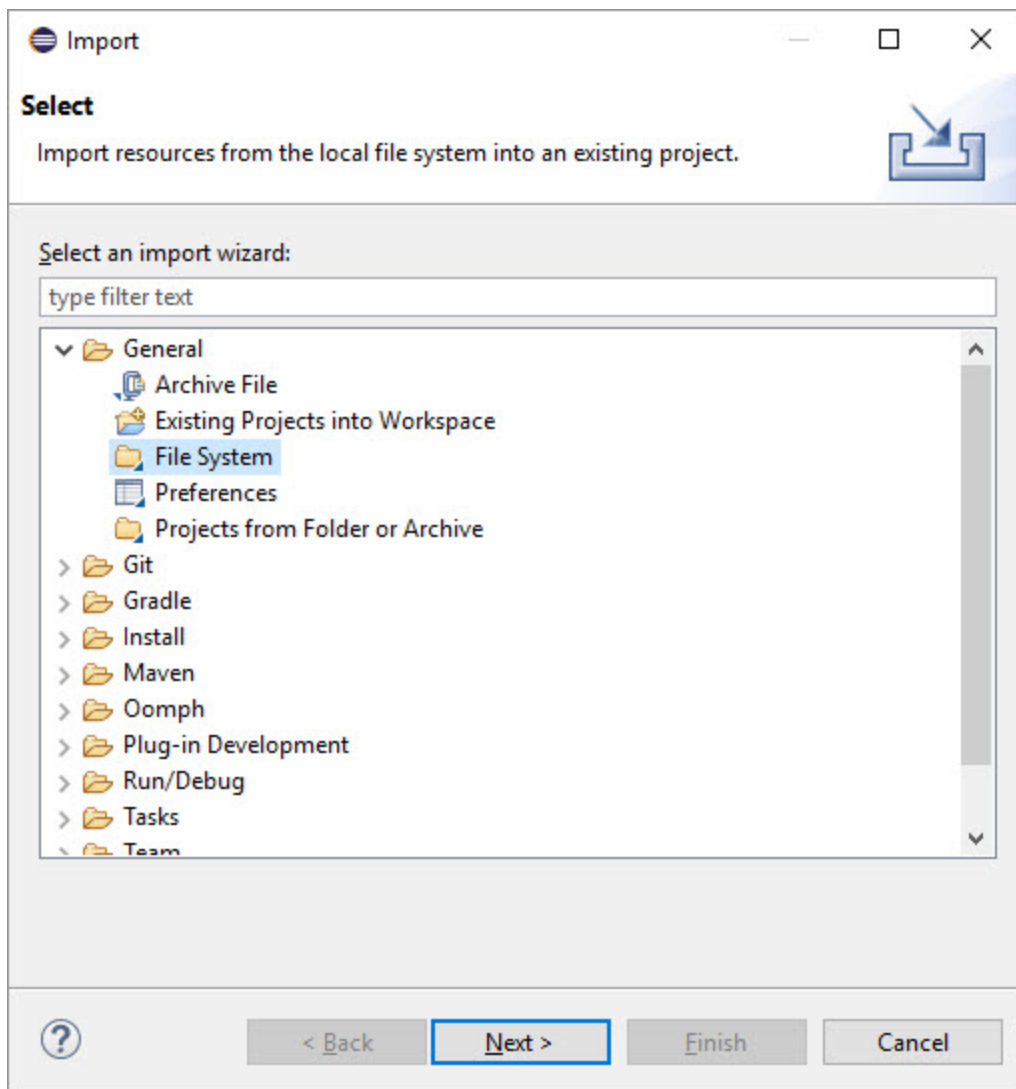
### *To import files:*

1. From the main menu, select **File > Import...** The **Import** dialog appears.

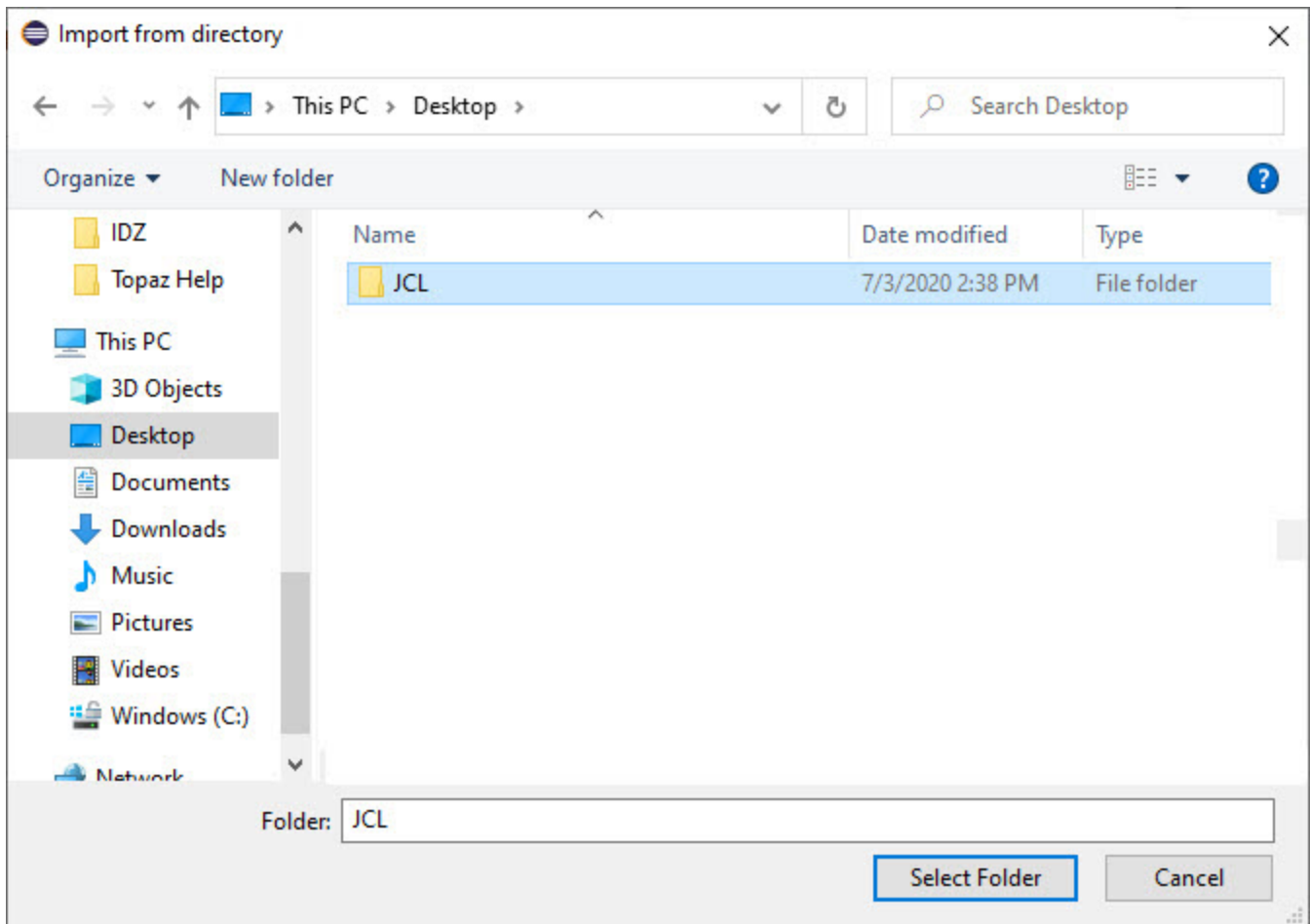


2. Expand the **General** folder, select **File System** and click **Next**.

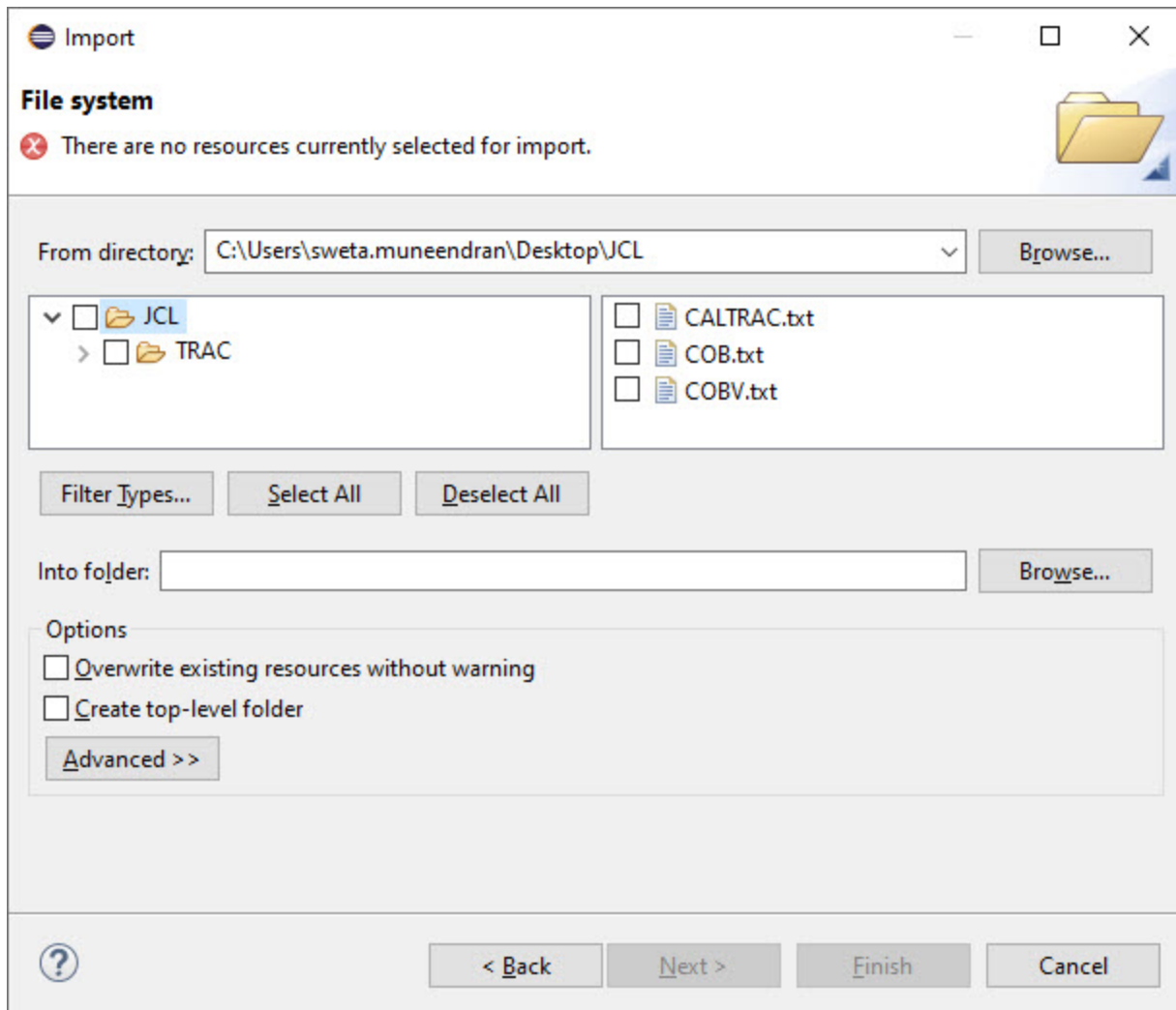




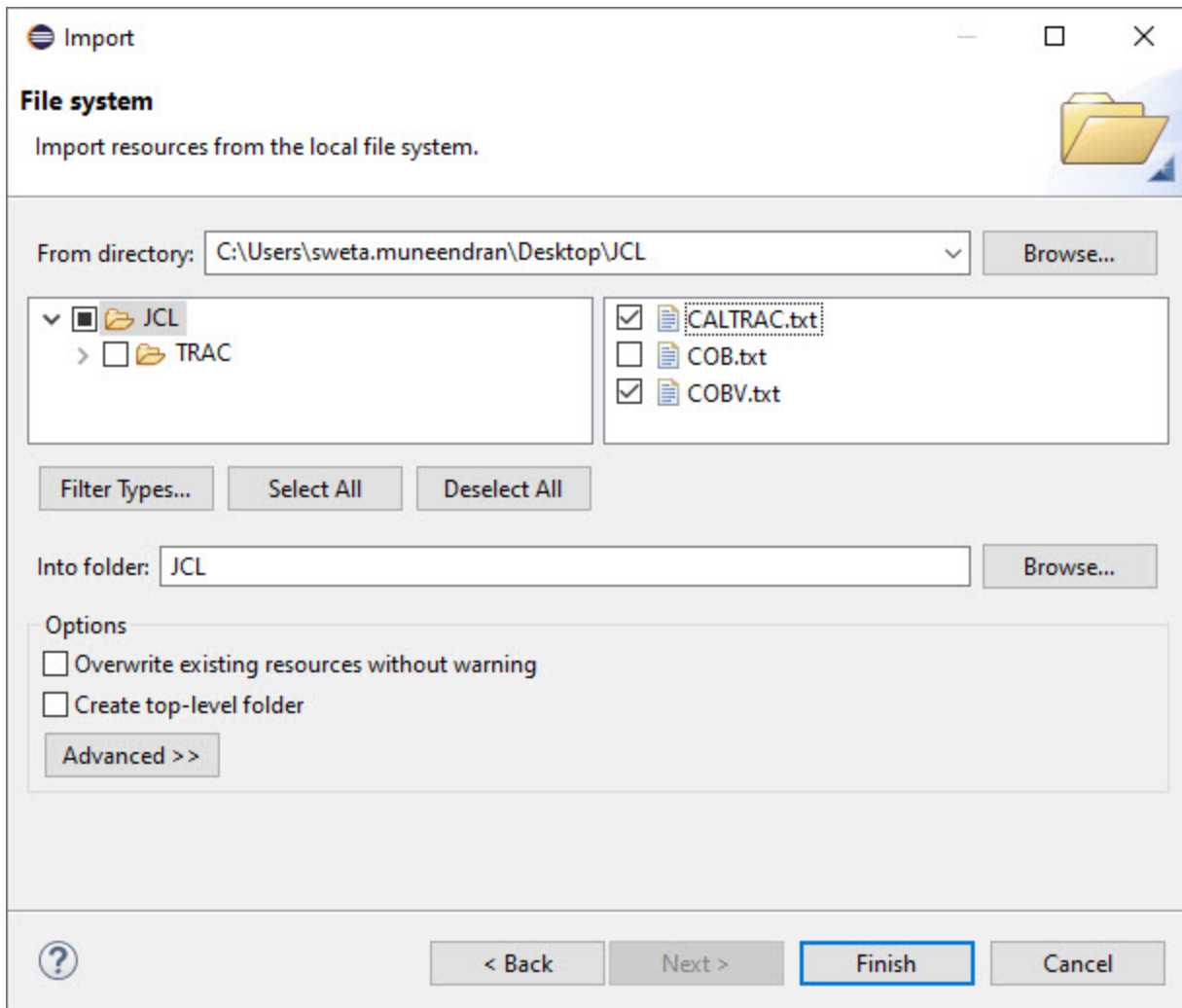
3. In the next page of the Import dialog, click the **Browse** button. The **Import from directory** dialog appears.



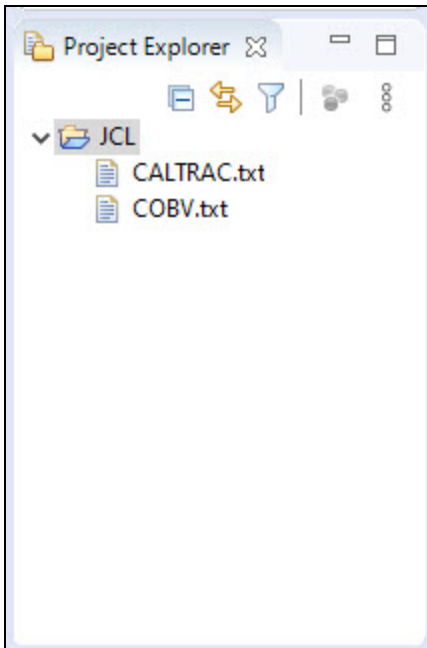
4. Select the directory from which you would like to add the files and click **Select Folder**. The directory is displayed in the left pane of the **Import** dialog.



5. To select exactly the files you want to add, expand the hierarchies in the left pane and select or clear the checkboxes that represent the folders in the selected directory. Then in the right pane, select or clear checkboxes for individual files.
6. In the **Into folder** option, specify the project or folder that will be the import destination.



7. When you have finished specifying your import options, click **Finish**. The files are displayed in the **Project Explorer** view under the specified destination.

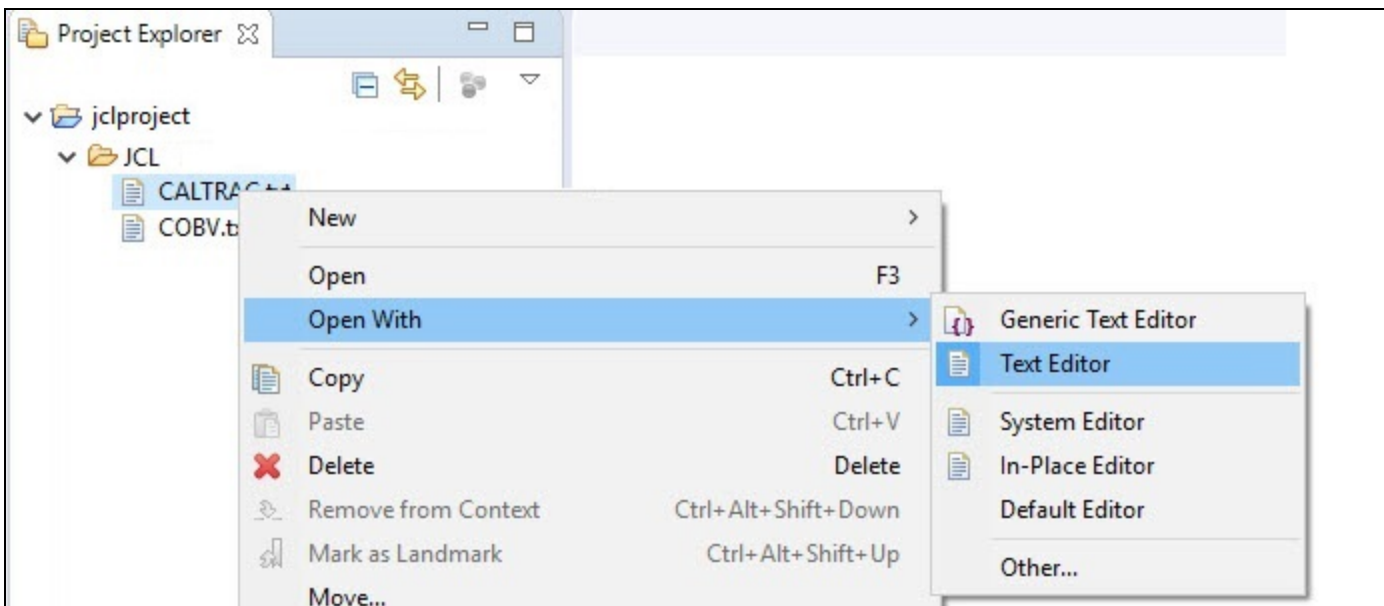


## Editing and Saving Local Files

The JCL editor allows you to edit and save a file that is in your local file system.

### To edit a file:

1. In the **Project Explorer** view, right-click on the file that you want to edit and select **Open With > Text Editor**.  
The file is opened in the JCL editor area.



2. After updating the file, do one of the following:
  - Press **Ctrl+ S**
  - Select **File > Save** or **Save As**
  - Right-click on the file and select **Save**.



- If the file opens in an external editor outside the Eclipse IDE, then you probably opened the file without selecting an editor.
- ASG recommends you to open the file in Text Editor.
- You may not see the JOB/SCAN functions if you have opened the file in different editor other than Text Editor.

See [Validating JCL](#) for information on scanning / validating a file.

# Validating JCL

ASG-JOB/SCAN validates JCL in an input job stream, so that potential errors, including run-time, hidden, site-specific and syntax errors can be corrected before the job stream is submitted for processing. Similarly, ASG-JOB/SCAN Plug-in processes the entire job stream in the z/OS environment and displays the results in JCL Scan and Problems views.

You can validate a JCL file using the following options:

- [Scan JCL](#)
- [Scan and submit JCL](#)


## Scan JCL

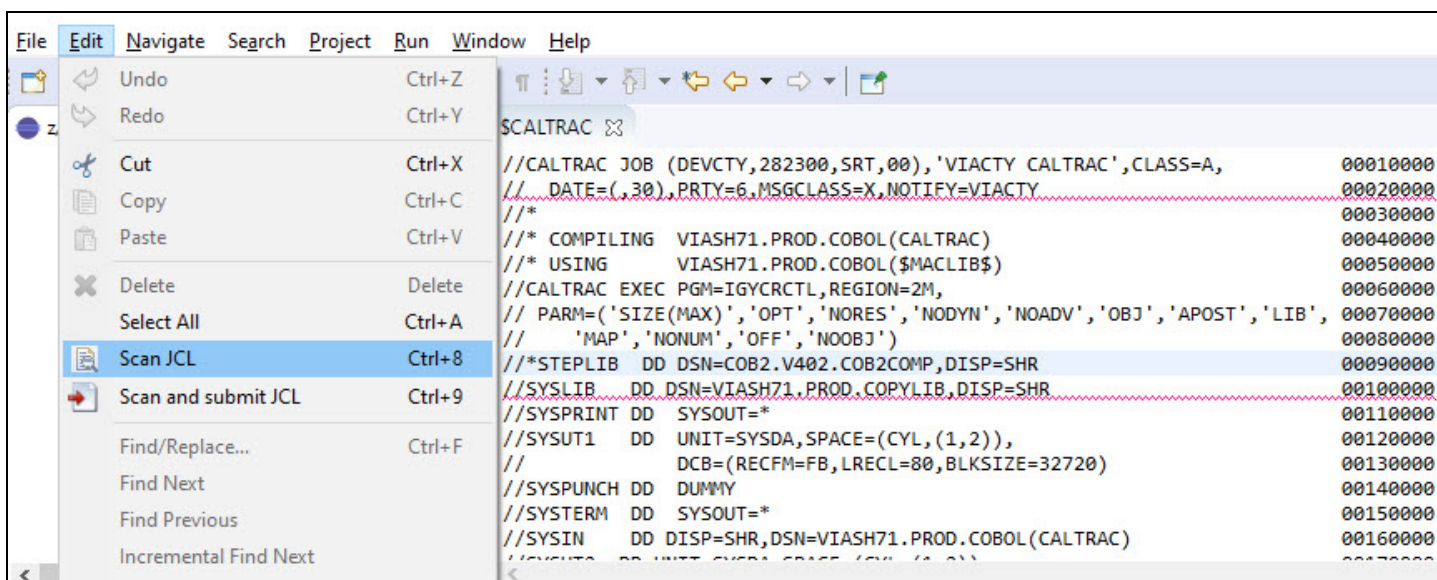
Scan JCL option available in the ASG-JOB/SCAN Plug-in is similar to ASG-JOB/SCAN's JEM and JSCAN edit macros. The edit macros validates the JCL file during an ISPF Edit session and displays the Structured JCL Listing (SJL) report. Similarly, the Scan JCL option validates the JCL file that is active in the JCL Editor and displays the results in JCL Scan and Problems views.



- If you are not connected to a JOB/SCAN server when you run the JOB/SCAN commands, you will be asked to connect to JOB/SCAN first before the commands are executed. Perform the steps in [Connecting to z/OS and ASG-JOB/SCAN](#) to connect to JOB/SCAN's server.
- When a file is opened from **Project Explorer** view, JOB/SCAN commands are available in JCL Editor **only**.
- JCL scan options and processing options must be specified before you begin to scan a JCL file. For more information on setting these options, see [Setting the JCL Scan Options](#) and [Setting the Processing Options](#)

### **To scan JCL:**

1. Open the file that you want to scan in JCL editor.
2. Make changes, if any, and save it.
3. Select **Scan JCL** option from the **Edit** menu or  icon from the toolbar.



4. JOB/SCAN Eclipse validates the input JCL and displays the reports in the **JCL Scan** and **Problems** views. For more information, refer to [Reports](#).



When you scan another file in the JCL editor, the report in the JCL Scan view is replaced. However, the messages in the Problems view are not replaced, instead they add up to the existing messages.



z/OS Navigator

CNTL ('DV060A.COBOL.CNTL')

- SBIND1
- SCALTRAC
- SCOBV42C
- SCOBV42L
- SCOBV42X
- SCOBV51C
- SCOBV51L
- SCOBV51X
- SCOBV52C
- SCOBV52L
- SCOBV52X
- SCOBV53C
- SCOBV53L
- SCOBV53X
- SCOBV61C
- SCOBV61L
- SCOBV61X
- SCOBV62A
- SCOBV62C

SCALTRAC

```
//CALTRAC JOB (DEVCTY,282300,SRT,00), 'VIACTY CALTRAC', CLASS=A,          00010000
//  DATE=(,30), PRTY=6, MSGCLASS=X, NOTIFY=VIACTY                       00020000
//*                                                                       00030000
//* COMPILING VIASH71.PROD.COBOL(CALTRAC)                               00040000
//* USING      VIASH71.PROD.COBOL($MACLIB$)                             00050000
//CALTRAC EXEC PGM=IGYCRCTL, REGION=2M,                                  00060000
// PARM=('SIZE(MAX)', 'OPT', 'NORES', 'NODYN', 'NOADV', 'OBJ', 'APOST', 'LIB', 00070000
//  'MAP', 'NONUM', 'OFF', 'NOOBJ')                                       00080000
//*STEPLIB DD DSN=COB2.V402.COB2COMP, DISP=SHR                          00090000
//SYSLIB DD DSN=VIASH71.PROD.COPYLIB, DISP=SHR                          00100000
//SYSPRINT DD SYSOUT=*                                                  00110000
//SYSUT1 DD UNIT=SYSDA, SPACE=(CYL, (1,2)),                             00120000
//      DCB=(RECFM=FB, LRECL=80, BLKSIZE=32720)                         00130000
//SYSPUNCH DD DUMMY                                                    00140000
//SYSTEM DD SYSOUT=*                                                    00150000
//SYSIN DD DISP=SHR, DSN=VIASH71.PROD.COBOL(CALTRAC)                   00160000
//SYST TN DD DUMMY                                                      00170000
```

Problems

2 errors, 0 warnings, 1 other

Description	Resource
> Errors (2 items)	
> Infos (1 item)	

JCL Scan

```
//CALTRAC JOB (DEVCTY,282300,SRT,00), 'VIACTY CALTRAC', CLASS=A,          00010000
//  DATE=(,30), PRTY=6, MSGCLASS=X, NOTIFY=VIACTY                       00020000
***ERROR - DSS1080E - INVALID JOB PARAMETER - "DATE ".
//*                                                                       00030000
//* COMPILING VIASH71.PROD.COBOL(CALTRAC)                               00040000
//* USING      VIASH71.PROD.COBOL($MACLIB$)                             00050000
//CALTRAC EXEC PGM=IGYCRCTL, REGION=2M,                                  00060000
// PARM=('SIZE(MAX)', 'OPT', 'NORES', 'NODYN', 'NOADV', 'OBJ', 'APOST', 'LIB', 00070000
//  'MAP', 'NONUM', 'OFF', 'NOOBJ')                                       00080000
//*STEPLIB DD DSN=COB2.V402.COB2COMP, DISP=SHR                          00090000
//SYSLIB DD DSN=VIASH71.PROD.COPYLIB, DISP=SHR                          00100000
***ERROR - DSS4050E - DATA SET NOT FOUND IN CATALOG
```

- After validation, JOB/SCAN Eclipse displays markers on the right and left margins of the JCL Editor. These markers help you to quickly identify a problem in a file. See [Markers](#) for more information.

SCALTRAC

```
//CALTRAC JOB (DEVCTY,282300,SRT,00), 'VIACTY CALTRAC', CLASS=A,          00010000
//  DATE=(,30), PRTY=6, MSGCLASS=X, NOTIFY=VIACTY                       00020000
//*                                                                       00030000
//* COMPILING VIASH71.PROD.COBOL(CALTRAC)                               00040000
//* USING      VIASH71.PROD.COBOL($MACLIB$)                             00050000
//CALTRAC EXEC PGM=IGYCRCTL, REGION=2M,                                  00060000
// PARM=('SIZE(MAX)', 'OPT', 'NORES', 'NODYN', 'NOADV', 'OBJ', 'APOST', 'LIB', 00070000
//  'MAP', 'NONUM', 'OFF', 'NOOBJ')                                       00080000
//*STEPLIB DD DSN=COB2.V402.COB2COMP, DISP=SHR                          00090000
//SYSLIB DD DSN=VIASH71.PROD.COPYLIB, DISP=SHR                          00100000
//SYSPRINT DD SYSOUT=*                                                  00110000
//SYSUT1 DD UNIT=SYSDA, SPACE=(CYL, (1,2)),                             00120000
//      DCB=(RECFM=FB, LRECL=80, BLKSIZE=32720)                         00130000
//SYSPUNCH DD DUMMY                                                    00140000
//SYSTEM DD SYSOUT=*                                                    00150000
//SYSIN DD DISP=SHR, DSN=VIASH71.PROD.COBOL(CALTRAC)                   00160000
//SYST TN DD DUMMY                                                      00170000
```


## Scan and submit JCL

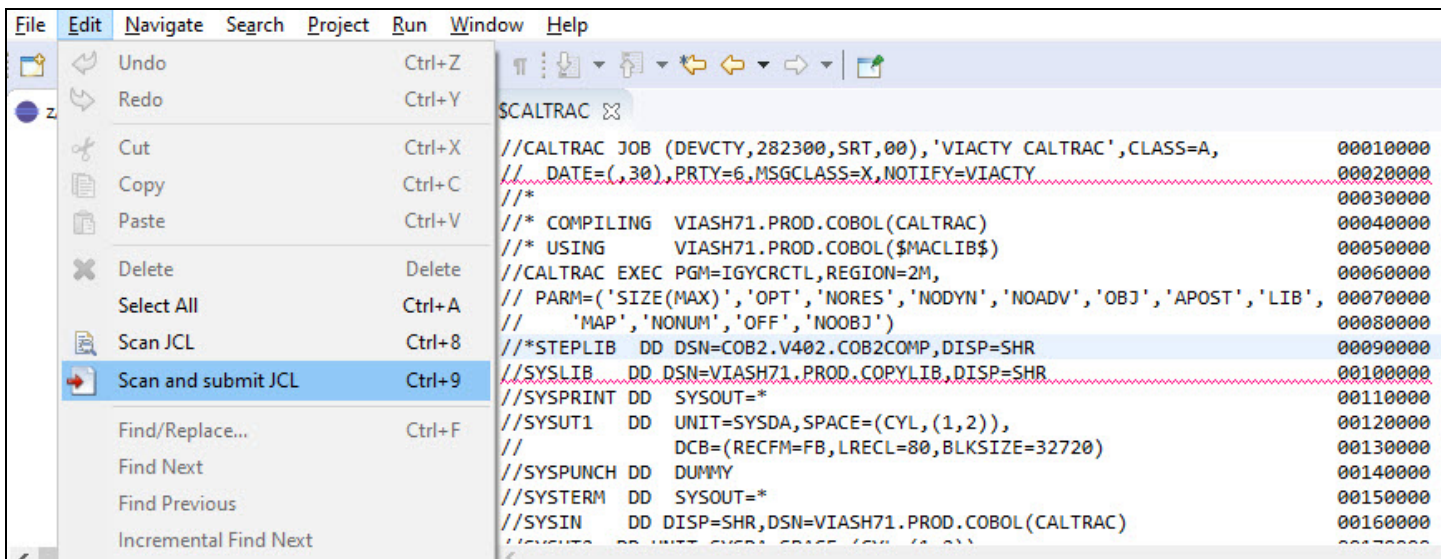
Scan and submit JCL option available in the ASG-JOB/SCAN Plug-in is similar to the ASG-JOB/SCAN's JSUB edit macro. The JSUB edit macro validates a JCL and submits the job from the edit session. Similarly, the 'Scan and submit JCL' option validates a file that is active in the JCL Editor and submits the job to z/OS. If the validation process completes with a severity level of 4 or less, the job is automatically submitted and no reports are displayed in the JCL Scan or Problems views. If the severity level is greater than 4, then the job is not submitted and the reports are displayed.



- If you are not connected to a JOB/SCAN server when you run the JOB/SCAN commands, you will be asked to connect to JOB/SCAN first before the commands are executed. Perform the steps in [Connecting to z/OS and ASG-JOB/SCAN](#) to connect to JOB/SCAN's server.
- When a file is opened from **Project Explorer** view, JOB/SCAN commands are available in JCL Editor **only**.
- JCL scan options and processing options must be specified before you begin to scan a JCL file. For more information on setting these options, see [Setting the JCL Scan Options](#) and [Setting the Processing Options](#)

### To scan and submit JCL:

1. Open the file that you want to scan in JCL Editor.
2. Make changes, if any, and save it.
3. Select **Scan and submit JCL** option from the **Edit** menu or  icon from the toolbar.



4. JOB/SCAN Eclipse validates the input JCL and displays the reports in the JCL Scan and Problems views depending on the severity. The job will be submitted if the severity is less than or equal to 4.

### Sample JCL file with severity level less or equal to 4

The below sample JCL file with severity level less than or equal to 4 is automatically submitted to z/OS and a success message is displayed in the JCL Scan and JOB/SCAN Console views.

The screenshot displays the z/OS Navigator interface. On the left, the 'CNTL' directory is expanded to show a list of JCL files, with '\$COBV42C' selected. The main pane shows the content of '\$COBV42C', which is a COBOL job definition. The job name is 'JOB (DEVCTY,282300,SRT,00)' and it is classified as 'COBOL 4.2 COMP'. The JCL includes various parameters such as 'NOTIFY=&SYSUID', 'MSGCLASS=X', 'REGION=0M', and 'TIME=20'. It also specifies the source file 'DV060A.NTIS' and the load library 'DV060A.COBV42'. The job is configured to execute the 'IGYCRCTL' program with various parameters like 'NOOPT', 'LIST', 'MAP', 'NONUM', 'NOOFF', 'S', 'LIST', 'MAP', 'S', 'NOADV', 'FASTSRT', 'RENT', 'TRUNC(OPT)', and 'NOXREF'. The job is submitted to the 'COB2.R32.COB2COMP' dataset with a shareable disposition (DISP=SHR). The JCL also includes several DD statements for datasets like 'COB2.R32.COB2COMP', 'SYSP.IGY122.SIGYCOMP', 'ESW.MVSCOB.IGY210.SIGYCOMP', 'SYSP.IGY331.SIGYCOMP', 'SYSP.IGY340.SIGYCOMP', 'SYSP.IGY410.SIGYCOMP', 'SYSP.IGY420.SIGYCOMP', 'CEE.SCEERUN', and 'CEE.SCEERUN2', all with a shareable disposition. The job is submitted to the 'SRC..COBOL' dataset with a shareable disposition.

Below the JCL content, the 'JCL Scan' view shows the message: 'Job submitted to z/OS.' The 'Console' view shows the following output:

```
JOB/SCAN Console
JCL severity level is 0. Job is ready to be submitted to z/OS.
Job submitted to z/OS successfully.
```

### Sample JCL file with severity level greater than 4

The below sample JCL file with severity level greater than 4 is not submitted and a failure message along with severity level is displayed in the JOB/SCAN Console view. Validation reports are displayed in JCL Scan and Problems views. For more information on reports, refer to [Reports](#).

The screenshot displays the z/OS Navigator interface with the JCL editor open to the file SCALTRAC. The JCL code includes job parameters like DATE, PRTY, MSGCLASS, NOTIFY, and various DD statements for compilation and execution. The Problems view at the bottom left indicates two errors and one other message. The Console view at the bottom right shows the job execution log, including an error message: '\*\*\*ERROR - DSS1000E - INVALID JOB PARAMETER - "DATE "'.



When you scan another file in the JCL editor, the report in the JCL Scan view is replaced. However, the messages in the Problems view are not replaced, instead they add up to the existing messages.

# Reports

ASG-JOB/SCAN Plug-in validates the input JCL for any errors, warnings and advisories (informational). The results of the validation are displayed in an easy-to-read report format. These reports are displayed automatically to the user in the **JCL Scan** and **Problems** views.

The reports contain the following messages:

## Error

These messages are likely to affect processing and should be corrected before the JCL is submitted.

## Warning

These messages may or may not affect processing and should be examined to determine if they must be corrected before the JCL is submitted.

## Advisory

These messages are informational only and are unlikely to affect processing.

## JCL Scan Report

In the JCL Scan report, the messages appear immediately after the JCL line that contains errors or warnings or advisories.

```

JCL Scan
//CALTRAC JOB (DEVCTY,282300,SRT,00),'VIACTY CALTRAC',CLASS=A,          00010000
// DATE=(,30),PRTY=6,MSGCLASS=X,NOTIFY=VIACTY                        00020000
***ERROR - DSS1080E - INVALID JOB PARAMETER - "DATE ".
//*                                                                    00030000
//* COMPILING VIASH71.PROD.COBOL(CALTRAC)                             00040000
//* USING      VIASH71.PROD.COBOL($MACLIB$)                           00050000
//CALTRAC EXEC PGM=IGYCRCTL,REGION=2M,                                00060000
// PARM=('SIZE(MAX)', 'OPT', 'NORES', 'NODYN', 'NOADV', 'OBJ', 'APOST', 'LIB', 00070000
//   'MAP', 'NONUM', 'OFF', 'NOOBJ')                                  00080000
//*STEPLIB DD DSN=COB2.V402.COB2COMP,DISP=SHR                          00090000
//SYSLIB DD DSN=VIASH71.PROD.COPYLIB,DISP=SHR                         00100000
***ERROR - DSS4050E - DATA SET NOT FOUND IN CATALOG
*ADVISORY - DSS8900A - DSN = "VIASH71.PROD.COPYLIB
//SYSPRINT DD SYSOUT="                                                00110000
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,2)),                             00120000
// DCB=(RECFM=FB.LRECL=80.BLKSIZE=32720)                             00130000

```

The messages are displayed in the following format:

[Message Type] - [Message Number] - [Message Text]

- Message types are:
  - Advisory
  - Error

- Warning
  - Message number is a five digit numeral prefixed with DSS and suffixed with message type.
  - Message text is a brief description of the problem.

After the messages, the report summarizes the Statistical Summary and the Space Analysis.

The Statistical Summary includes the total number of PROGRAM(S), PROC(S), INCLUDE(S), DD STATEMENT(S) and input / output TAPE DD STATEMENT(S) that are in the scanned JCL file.

```

JCL Scan
*ADVISORY - DSS8940A - TEMPORARY DATA SET WILL BE USED.
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,2))           00180000
*ADVISORY - DSS8940A - TEMPORARY DATA SET WILL BE USED.
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(1,2))           00190000
*ADVISORY - DSS8940A - TEMPORARY DATA SET WILL BE USED.
//SYSUT5 DD UNIT=SYSDA,SPACE=(CYL,(1,2))           00200000
*ADVISORY - DSS8940A - TEMPORARY DATA SET WILL BE USED.
//SYSUT6 DD UNIT=SYSDA,SPACE=(CYL,(1,2))           00210000
*ADVISORY - DSS8940A - TEMPORARY DATA SET WILL BE USED.
//SYSUT7 DD UNIT=SYSDA,SPACE=(CYL,(1,2))           00220000
*ADVISORY - DSS8940A - TEMPORARY DATA SET WILL BE USED.
//SYSLIN DD DUMMY                                   00230000
//*                                                  00240000
***ERROR - DSS2050E - PROGRAM "TGYCRCTI" NOT FOUND
JOB (CALTRAC ) CONSISTS OF      0 PROCS,      1 PROGRAMS,      13 DD STATEMENTS, AND      24 TOTAL RECORDS.
      THERE ARE      0 INPUT AND      0 OUTPUT TAPE DD STATEMENTS.
      SPACE ANALYSIS (DISK ) TEMP CYL      7 -PRIMARY      14 -SEC
  
```

The Space Analysis includes list of all new space requirements for the validated job stream. The total allocation requirements are broken down by UNIT and VOLSER, and indicate whether a file is permanent or temporary.

Next, the ERROR SUMMARY AND COUNTS section summarizes the total number of errors, warnings, and advisories encountered after scanning the JCL. This section is displayed only when you select the **Include summary in JCL scan output** option in the **JCL Scan Options** tab in the **JOB/SCAN** preference page.

```

JCL Scan
*** ERROR SUMMARY AND COUNTS ***
 9 ADVISORY LEVEL
 0 WARNING LEVEL
 3 ERROR LEVEL
12 TOTAL ISSUED          0 SUPPRESSED.
  THE HIGHEST SEVERITY CODE ENCOUNTERED WAS      8

Options in Effect
-----
RUN OPTIONS
  JLJPC PROCLIB TYPE=PDS
  USE ERROR TABLE =NO
  PROCESSING SPECIFICATIONS =NOCONDCODE
  CATALOG ACCESS =YES,VTOC,PDSMEM,PGM
  PROGRAM PRODUCT INTERFACE =IDCAMS
  SCHEDULER =
  ACCESS CONTROL CARDS =YES,LIMIT=0
  SUBSYSTEM =NONE
  JOB LIMIT =0
  SUPPRESS ERROR MESSAGES =NO
REPORT OPTIONS
  STRUCTURED JCL LISTING=PRINT,ERRORTXT=SHORT
INPUT SELECTION
  DDNAME=JCLIN
  SOURCE TYPE=SEQ
FILE ALLOCATIONS
  JLJPC=SHR,DSN=DV405A.JCL

```

At the end is the Options in Effect section. This section summarizes the JCL scan options and the processing options that are used in validating a JCL. This section is displayed only when you select the **Include summary in JCL scan output** option in the **JCL Scan Options** tab in the **JOB/SCAN** preference page.

## Problems Report

The Problems report displays the list of messages that are identified after a JCL file is validated.

At the top, the report summarizes the number of errors, warnings, and advisories encountered. Subsequent section displays the list of messages in each category.

Description	Resource	Path	Location	Type
>  Errors (2 items)				
>  Infos (100 of 401 items)				

Expand a category by clicking the icon to view the list.

Description	Resource	Path	Location	Type
Errors (2 items)				
DATA SET NOT FOUND IN CATALOG	SCALTRAC	/jclproject/DV060A....	line 10	JCL Scan Mes...
INVALID JOB PARAMETER - "DATE ".	SCALTRAC	/jclproject/DV060A....	line 2	JCL Scan Mes...
Infos (100 of 401 items)				
DSN = "VIASH71.PROD.COPYLIB ""},{"error-id	SCALTRAC	/jclproject/DV060A....	line 10	JCL Scan Mes...
TEMPORARY DATA SET WILL BE USED.	SCOBV42C	/jclproject/DV060A....	line 65	JCL Scan Mes...
TEMPORARY DATA SET WILL BE USED.	SCOBV42C	/jclproject/DV060A....	line 65	JCL Scan Mes...
TEMPORARY DATA SET WILL BE USED.	SCOBV42C	/jclproject/DV060A....	line 65	JCL Scan Mes...
TEMPORARY DATA SET WILL BE USED.	SCOBV42C	/jclproject/DV060A....	line 65	JCL Scan Mes...

Information about the messages are listed in a table format containing the following columns.

Column	Description
Description	Message describing the problem
Resource	Member name in PDS or name of sequential dataset
Path	Path of the Partitioned dataset name or sequential dataset
Location	Indicates the JCL line number where the error appears
Type	<i>JCL Scan Message</i>

The Problems view offers a simple and quick way to locate the problem line that is associated with the message.

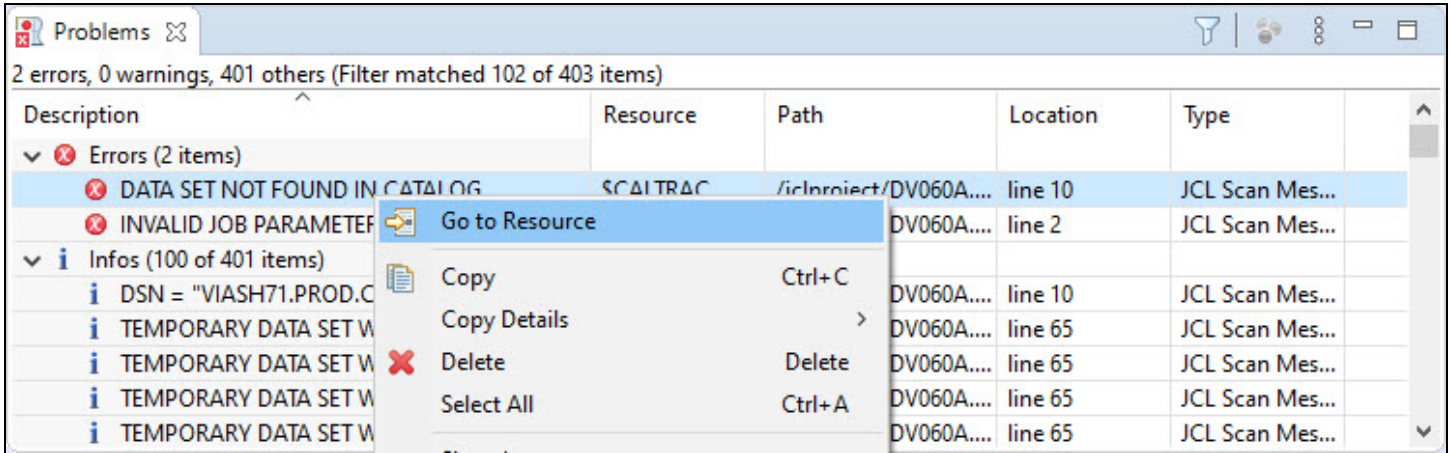
You can locate the problem either by:

- double-clicking on the message



Or

- right-click on the message to bring up the problems view's contextual menu and select the **Go to Resource** option.



# Markers

Markers are small indicators that appear on the right and left margins of the JCL editor after a file is validated. These indicators enable you to quickly identify problems in your file. With these markers, user can navigate between the problem lines and identify the highest severity in a file.

Markers are categorized as follows:

- Errors
- Warnings
- Advisories (Information)



By default, only the error markers are displayed in the right and left margins. You can choose to show / hide markers as per your preference. See [Show or Hide Markers](#) for more information.

## Left-margin Markers

Markers on the left-margin are placed next to the lines that are associated with some problems. These markers move with the source lines when the content in the JCL editor is scrolled.

The left-margin can have different icons with the following meaning:

Icon	Description
	Errors
	Warnings
	Advisories

Markers in left-margin

958

When you hover the mouse cursor over a marker, a pop-up shows the description of the problem(s) associated with the line.

```

XMIT2AKR
//XMIT2AKR JOB (DEVCTY,282300,SRT,00),'DUMP XMIT',CLASS=A,MSGCLASS=X, 00010000
//      NOTIFY=&SYSUID,TIME=(4,30),PRTY=6,REGION=4096K 00020000
//* 00030000
//DUMP EXEC PGM=DUMPXMIT,PARM=' LOGICAL,TERSE' 00040000
//STEPLIB DD DSN=DV060A.DEVL.LOAD,DISP=SHR 00050000
Data set 'CUSTDUMP.T768917.XMI' was not found in catalog.R 00060000
//SYSPRINT DD SYSOUT=* 00070000
//SYSOUT DD DSN=DV060A.HUGE.AKR,DISP=SHR 00080000
//*YSOUT DD DSN=DV060A.TEST.HEXDUMP.DUMPXMIT,DISP=(NEW,CATLG), 00090000
//* UNIT=SRTDA,SPACE=(CYL,(1,1)) 00100000
//*

```

## Right-margin Markers

Markers on the right-margin show the relative locations of the tagged lines in the file. As a result, the markers are still visible when the margin shrinks to fit the screen. These markers do not move when the content in the editor is scrolled.

Instead, clicking any marker in the right-margin would highlight the line associated with that marker. This allows you to quickly step through the problem lines in the editor by simply clicking the markers one by one.

Markers on the right-margin are color coded according to the message type:

Color	Description
Blue	Advisory messages
Yellow	Warning messages
Red	Error messages

```

XMIT2AKR
//XMIT2AKR JOB (DEVCTY,282300,SRT,00),'DUMP XMIT',CLASS=A,MSGCLASS=X, 00010000
//      NOTIFY=&SYSUID,TIME=(4,30),PRTY=6,REGION=4096K 00020000
//* 00030000
//DUMP EXEC PGM=DUMPXMIT,PARM=' LOGICAL,TERSE' 00040000
//STEPLIB DD DSN=DV060A.DEVL.LOAD,DISP=SHR 00050000
//SYSIN DD DSN=CUSTDUMP.T768917.XMI,DISP=SHR 00060000
//SYSPRINT DD SYSOUT=* 00070000
//SYSOUT DD DSN=DV060A.HUGE.AKR,DISP=SHR 00080000
//*YSOUT DD DSN=DV060A.TEST.HEXDUMP.DUMPXMIT,DISP=(NEW,CATLG), 00090000
//* UNIT=SRTDA,SPACE=(CYL,(1,1)) 00100000
//*

```

When you hover the mouse cursor over a marker, a pop-up shows the description of the problem(s) associated with the line.

```

XMIT2AKR
//XMIT2AKR JOB (DEVCTY,282300,SRT,00),'DUMP XMIT',CLASS=A,MSGCLASS=X, 00010000
//      NOTIFY=&SYSUID,TIME=(4,30),PRTY=6,REGION=4096K 00020000
//* 00030000
//DUMP EXEC PGM=DUMPXMIT,PARM='LOGICAL,TERSE' 00040000
//STEPLIB DD DSN=DV060A.DEVL.LOAD,DISP=SHR 00050000
//SYSIN DD DSN=CUSTDUMP.T768917.XMI,DISP=SHR 00060000
//SYSPRINT DD SYSOUT=* 00070000
//SYSOUT DD DSN=DV060A.HUGE.AKR,DISP=SHR 00080000
//*YSOUT DD DSN=DV060A.TEST.HEXDUMP.DUMPXMIT,DISP=(NEW,CATLG), 00090000
//* UNIT=SRTDA,SPACE=(CYL,(1,1)) 00100000
//*

```

A tooltip is visible over the yellow warning icon on the line for DSN=DV060A.HUGE.AKR, containing the text: "Data set 'DV060A.HUGE.AKR' has been migrated."

The square that appears at the top right-margin shows the highest severity found in the file. Messages have the following severity levels:

Severity	Color	Description
High	Red	Error messages
Medium	Yellow	Warning messages
Low	Blue	Advisory messages

The square is not displayed if there are no messages associated with the file.

When you hover the mouse cursor over the square, a pop-up shows the total number of severities encountered in the current file.

```

XMIT2AKR
//XMIT2AKR JOB (DEVCTY,282300,SRT,00),'DUMP XMIT',CLASS=A,MSGCLASS=X, 00010000
//      NOTIFY=&SYSUID,TIME=(4,30),PRTY=6,REGION=4096K 00020000
//* 00030000
//DUMP EXEC PGM=DUMPXMIT,PARM='LOGICAL,TERSE' 00040000
//STEPLIB DD DSN=DV060A.DEVL.LOAD,DISP=SHR 00050000
//SYSIN DD DSN=CUSTDUMP.T768917.XMI,DISP=SHR 00060000
//SYSPRINT DD SYSOUT=* 00070000
//SYSOUT DD DSN=DV060A.HUGE.AKR,DISP=SHR 00080000
//*YSOUT DD DSN=DV060A.TEST.HEXDUMP.DUMPXMIT,DISP=(NEW,CATLG), 00090000
//* UNIT=SRTDA,SPACE=(CYL,(1,1)) 00100000
//*

```

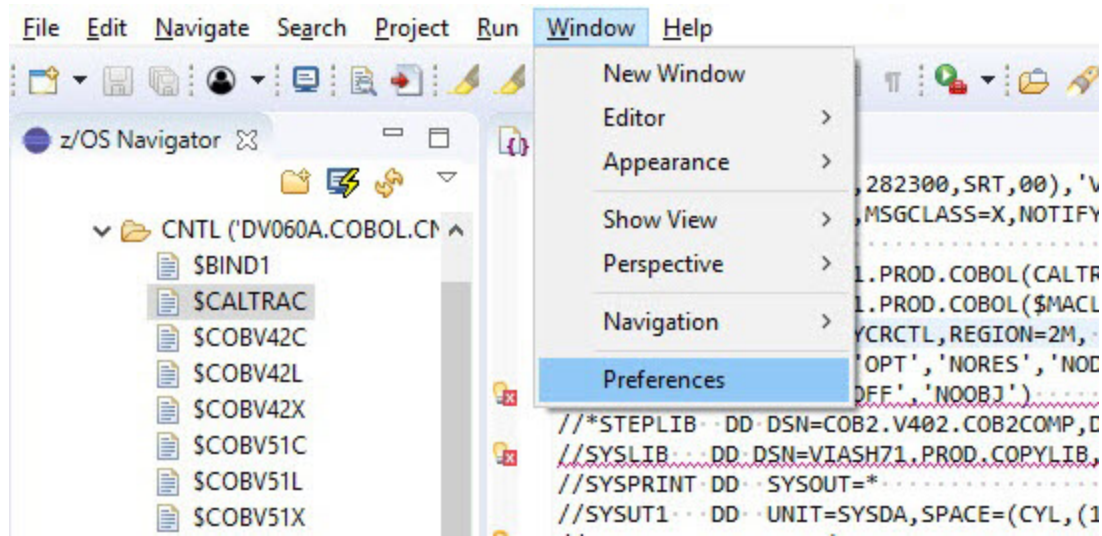
A tooltip is visible over the yellow warning icon, containing the text: "Errors: 1 Warnings: 1"

## Show or Hide Markers

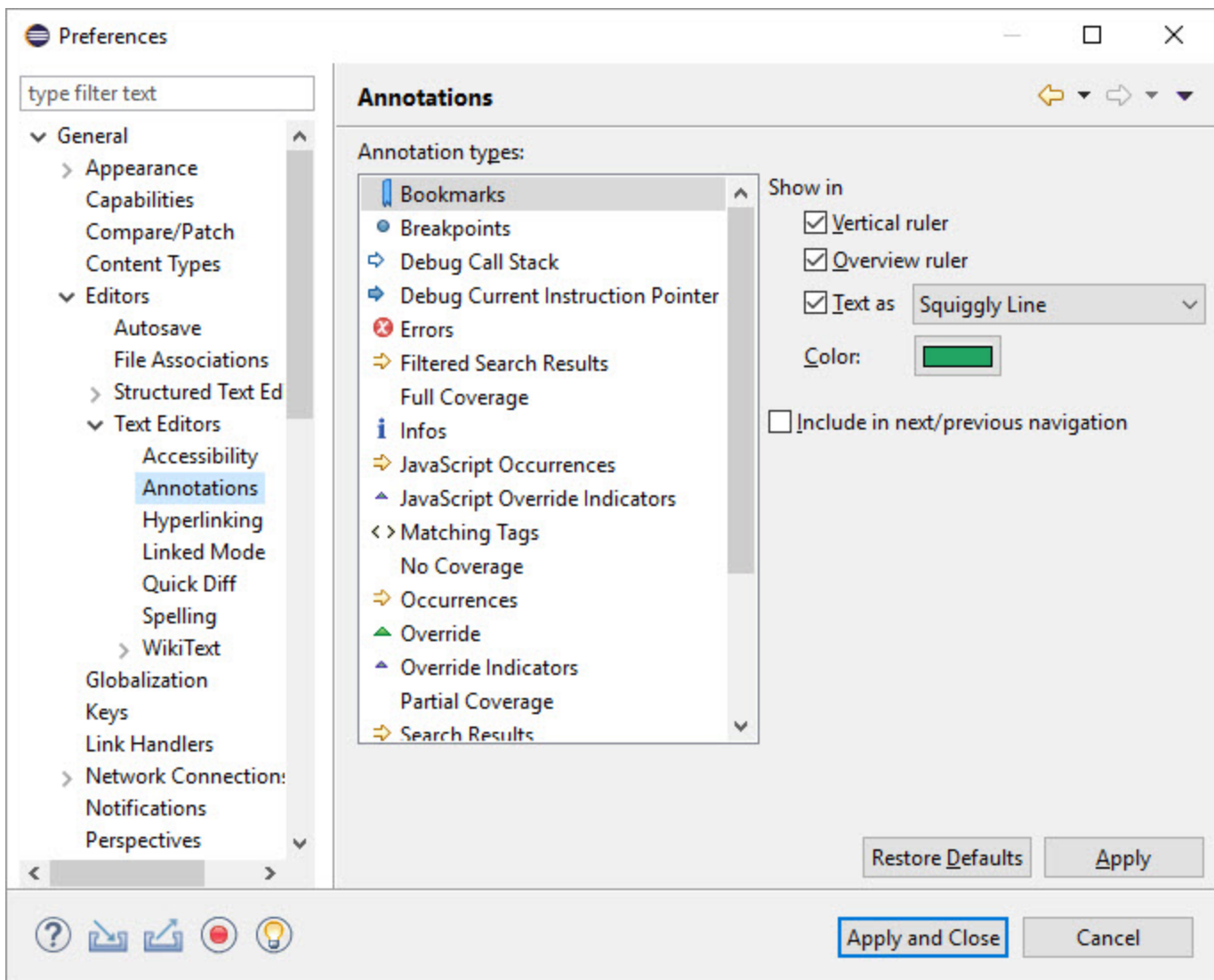
Markers like errors, warnings and infos are used to quickly identify the problems in a file. You might find these markers useful when you are validating a JCL file. You may choose how you want these markers to be displayed through the Preferences dialog.

### To show or hide markers:

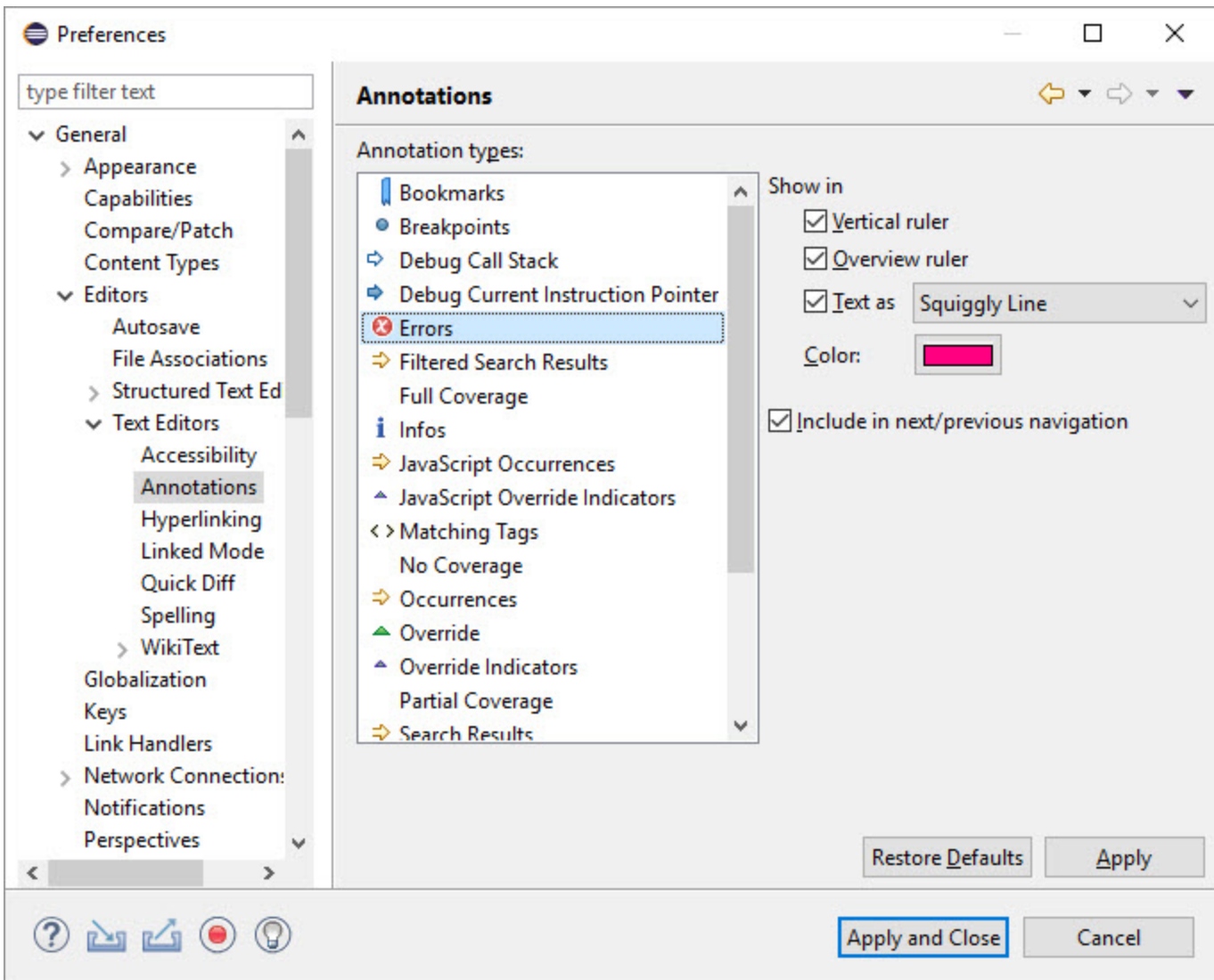
1. From the **Window** menu, select **Preferences**.



2. In the **Preferences** dialog, select **General > Editors > Text Editors > Annotations**.



3. In the **Annotations** page, under the **Annotations types**, select **Errors / Infos / Warnings** as per your requirement.



4. Under **Show in**, select or deselect the following options:

Options	Description
Vertical ruler	This option control the markers on the left-margin.
Overview ruler	This option control the markers on the right-margin.
Text as	This option control the format of the tagged lines in the JCL editor region. You can select on of the following formats: <ul style="list-style-type: none"> <li>• Box</li> <li>• Dashed Box</li> <li>• Highlighted</li> <li>• Native Problem Underline</li> </ul>

Options	Description
	<ul style="list-style-type: none"><li>• Squiggly Line</li><li>• Underlined</li><li>• Vertical Bar</li></ul>

5. After choosing your preference, click **Apply** to save the changes or click **Apply and Close** to save the changes and close the **Annotations** dialog.



# Keyboard Shortcuts

In this topic you can find the list of all the keyboard shortcuts for different tasks performed in ASG-JOB/SCAN Plug-in. Also, the topic explains the procedure to customize the default shortcuts.

The following table shows the shortcut keys used in ASG-JOB/SCAN Plug-in:

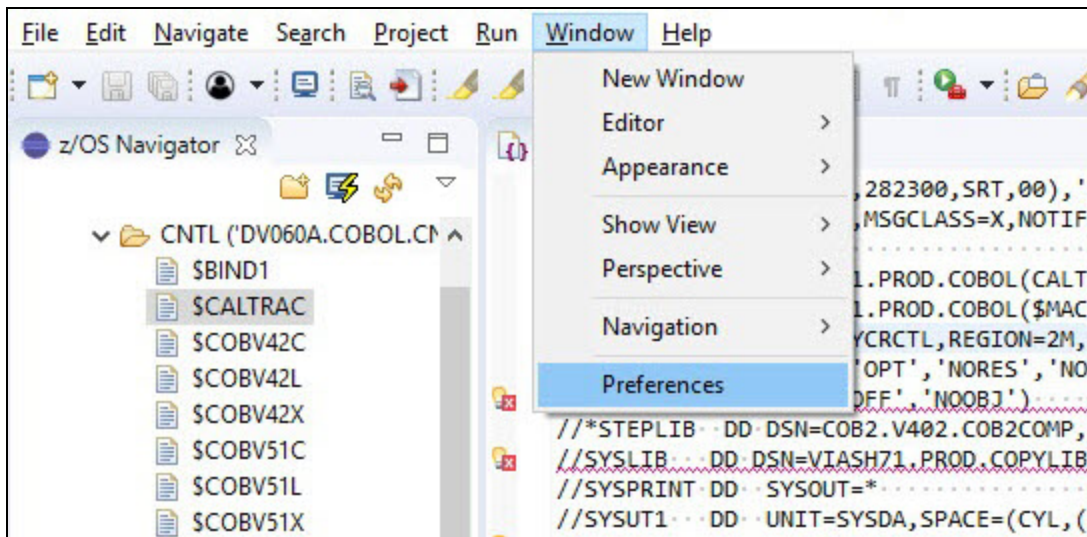
Task	Default Shortcut
Scan JCL	Ctrl + 8
Scan and submit JCL	Ctrl + 9
Save to z/OS	Ctrl + 5
Save to z/OS As...	Ctrl + Shift + 5

## Customizing Keyboard Shortcuts

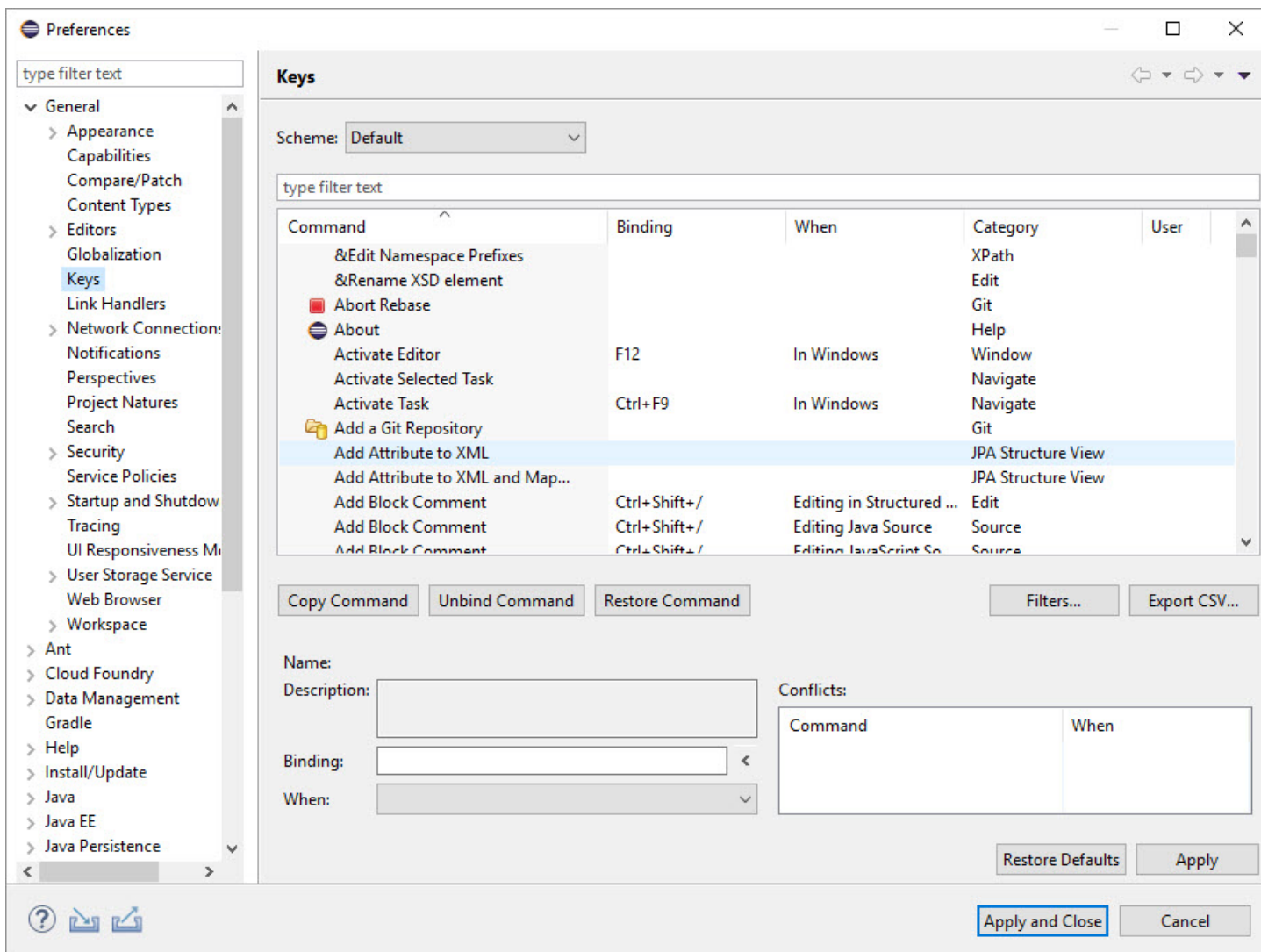
The key combinations for the existing shortcuts can be disabled or modified as required.

**To customize shortcuts:**

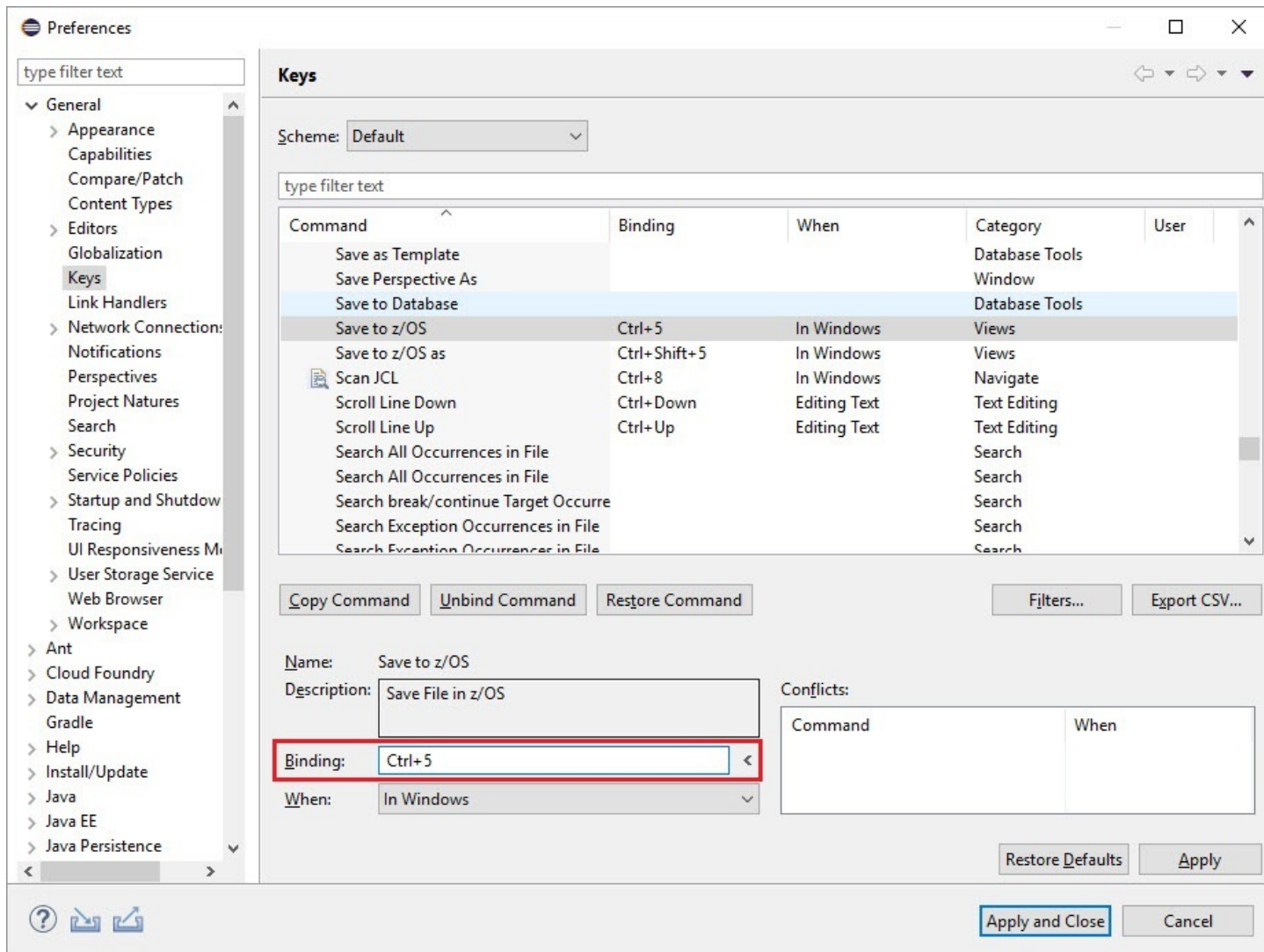
1. From the **Window** menu, select **Preferences**.



2. In the **Preferences** dialog, select **General > Keys**.



- In the **Keys** page, select the **Command** that you like to change. The values associated with the command is populated in the fields.



- To change the key combinations, edit the **Binding** field.
- If required, you can edit the values in the other fields.
- click **Apply** to save the changes or click **Apply and Close** to save the changes and close the **Preferences** dialog.

# ASG-JOB/SCAN REST API Documentation

The ASG-JOB/SCAN REST API service is used to interact with ASG-JOB/SCAN REST server. It gives the ability to scan the JCLs from any application. The ASG-JOB/SCAN REST API service is defined using the OpenAPI 3.0 specification.

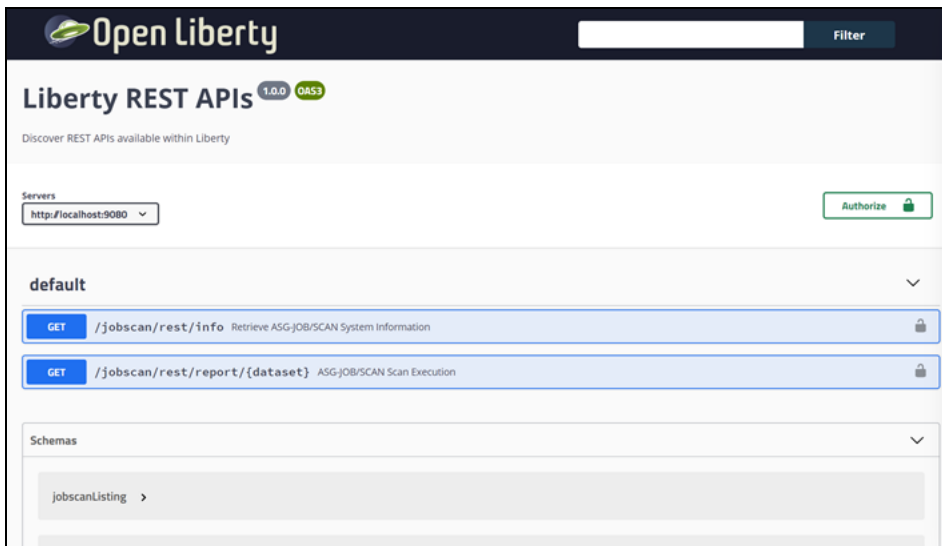
The ASG-JOB/SCAN REST API user interface is accessible at the URL: `http://<Host_Name>:<Port>/api/explorer/`

wherein:

*Host\_Name* indicates the host name or IP address of the z/OS system

*Port* indicates the port number of the host machine on which the JOB/SCAN REST API service is running

Example: `http://mvssyse.asg.com:60080/api/explorer/`

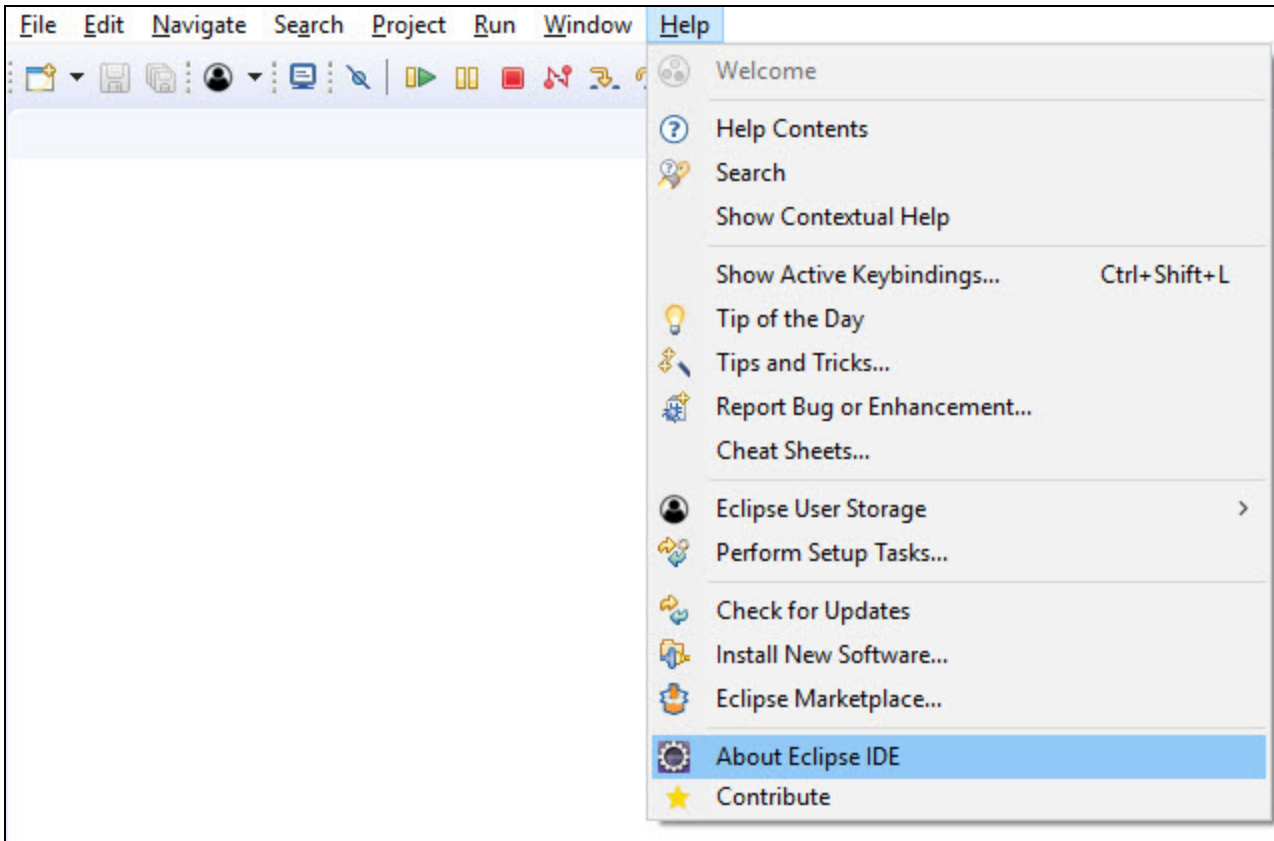


# Uninstallation

Before installing a new version of ASG-JOB/SCAN Plug-in, it is recommended to remove any prior versions from the server.

## To uninstall ASG-JOB/SCAN Plug-in:

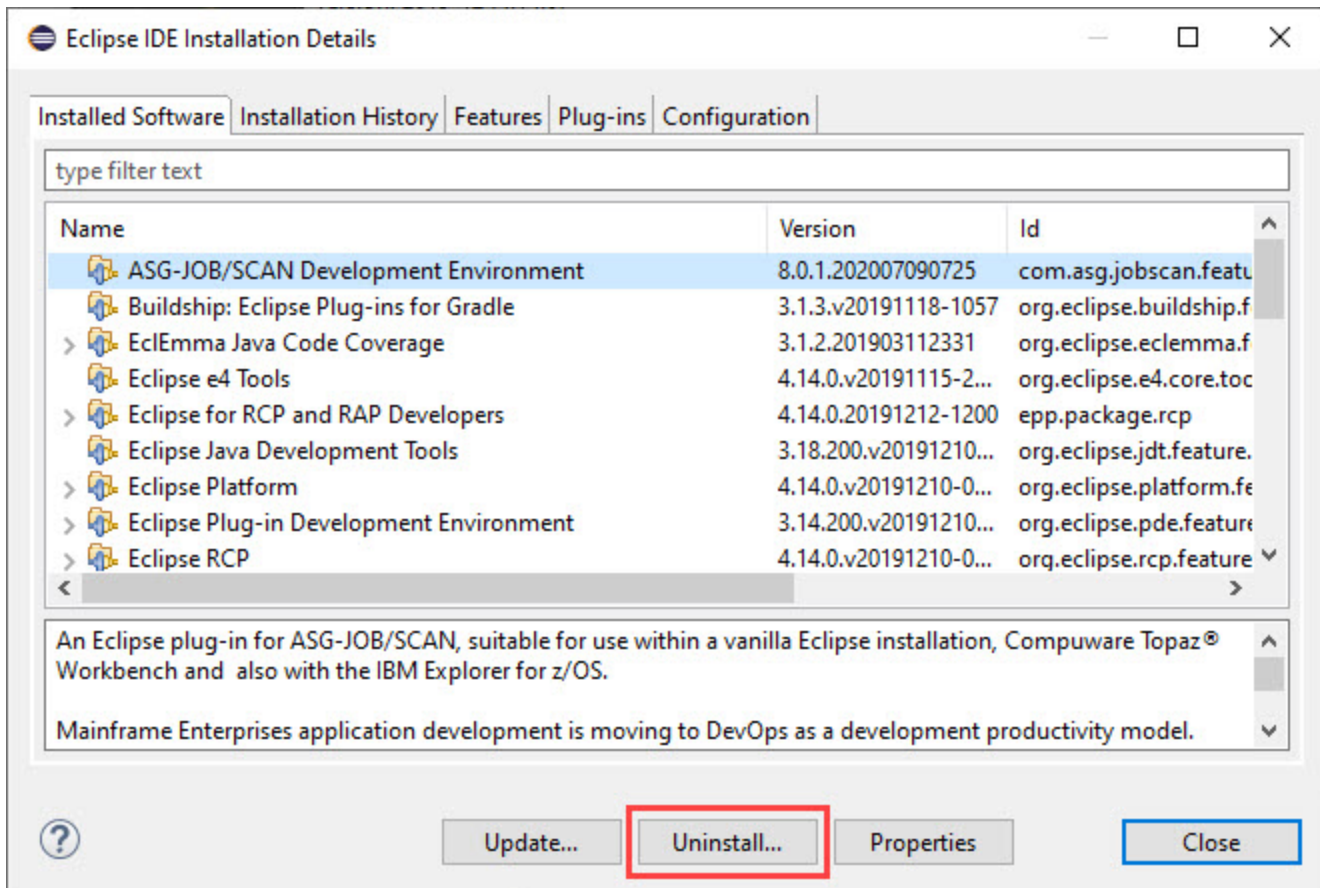
1. Start Eclipse.
2. From the menu, select **Help > About Eclipse IDE**.



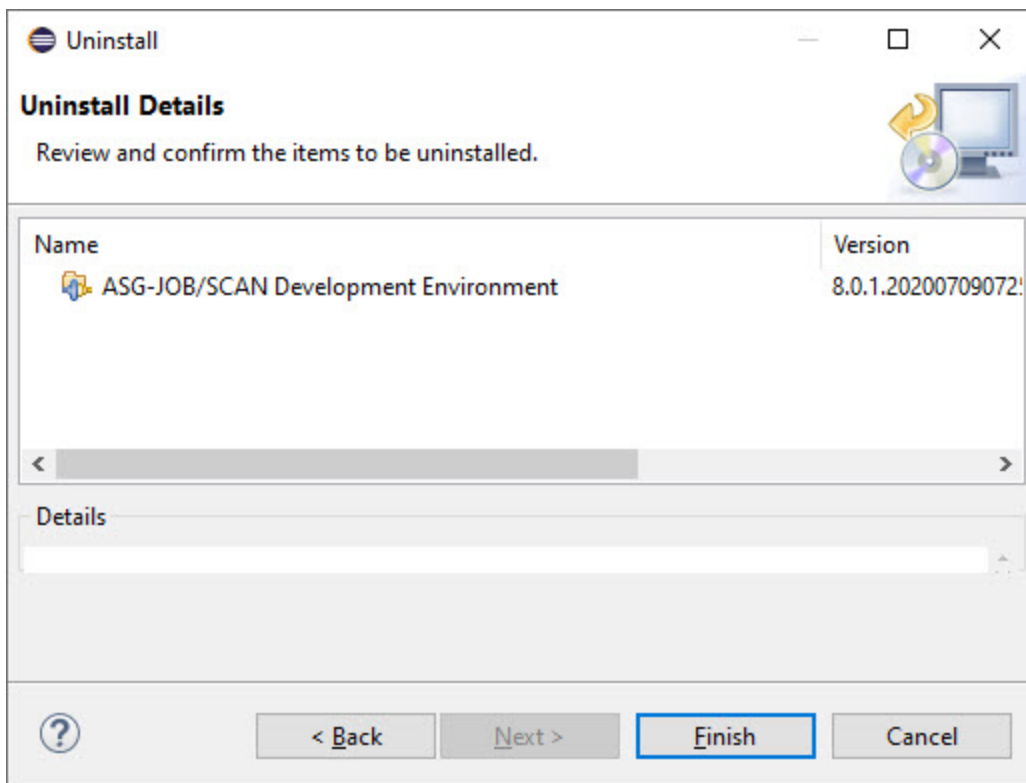
3. In the **About Eclipse IDE** dialog, click **Installation Details**.



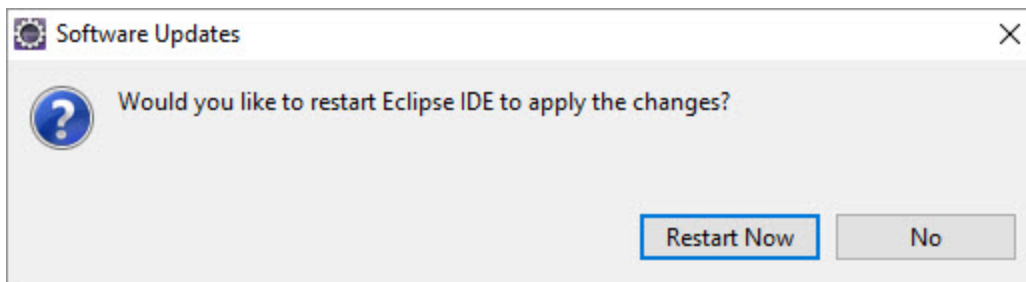
4. In the **Installed Software** tab, select **ASG-JOB/SCAN Development Environment** and click **Uninstall...**



5. Review the uninstall items and click **Finish**.



6. When the uninstallation is complete, you are prompted to restart the Eclipse application. Click **Restart Now**.



**To verify if the plug-in has been uninstalled:**

1. Start Eclipse
2. From the menu, select **Help > About Eclipse IDE**.
3. In the **About Eclipse IDE** dialog, click **Installation Details**.
4. In the **Installed Software** tab, you should not see **ASG-JOB/SCAN Development Environment** in the list of installed software. This indicates that ASG-JOB/SCAN Plug-in has been uninstalled.



# ASG-JOB/SCAN Plug-in for IBM Explorer for z/OS (z/OS Explorer)

This documentation introduces the features and benefits of ASG-JOB/SCAN Plug-in, explains how to install, and execute the product in IBM Explorer for z/OS environment.

The documentation consist of following topics:

- [Introduction](#)
- [Installation](#)
- [User Interface](#)
- [Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator](#)
- [Setting the JCL Scan Options](#)
- [Setting the Processing Options](#)
- [Connecting to z/OS from Remote Systems](#)
- [Working with z/OS Datasets](#)
- [Working with Local Files](#)
- [Validating JCL](#)
- [Reports](#)
- [Markers](#)
- [Keyboard Shortcuts](#)
- [ASG-JOB/SCAN REST API Documentation](#)
- [Uninstallation](#)

# Introduction

## What is JOB/SCAN?

JOB/SCAN is a comprehensive Job Control Language (JCL) management tool that provides JCL validation, reformatting, and standards enforcement. It is designed to help developers operate a streamlined JCL environment on the z/OS mainframe.

## What is IBM Explorer for z/OS?

IBM Explorer for z/OS (z/OS Explorer) is an Eclipse-based Integrated Development Environment (IDE) that supports end-to-end development of z/OS applications. z/OS Explorer consists of an integrated set of tools that helps programmers in developing and debugging programs designed for z/OS environments.

## What is JOB/SCAN Plug-in for IBM Explorer for z/OS?

JOB/SCAN Plug-in is an Eclipse plug-in that strategically integrates the mainframe capabilities of ASG-JOB/SCAN into a developer's DevOps user interface. IBM Explorer for z/OS (z/OS Explorer) is one of the preferred development environments for all z/OS applications. ASG extends JOB/SCAN Plug-in's JCL validation capability to z/OS Explorer, thereby complementing and strengthening its support for JCL development. The plug-in provides JCL management tasks with mainframe application workflow in an intuitive user interface. The simple interface provides developers to seamlessly initiate JCL integrity, verification, and reporting.



The JOB/SCAN Plug-in is a separately licensed option of ASG-JOB/SCAN. You need a license key to access it. Contact [ASG Customer Support](#) to obtain a license.

## Features of JOB/SCAN Plug-in:

- Offers a modern and easy-to-use User Interface (UI) to interact with the z/OS system.
- Evaluates JCL for any syntax errors.
- Enforces site-specific JCL standards.
- Submits jobs to the mainframe for execution.
- Provides effective reporting facility.
- Includes RESTful web services API to automate ASG-JOB/SCAN scans in a DevOps toolchain environment.

This document assumes that the user:

- has installed ASG-JOB/SCAN and running on a supported z/OS system.
- has authorization to access resources on the z/OS system.
- is familiar with the functions implemented in ASG-JOB/SCAN. If you are new to ASG-JOB/SCAN, refer to [ASG-JOB/SCAN® 8.0.0 Documentation](#).

# Installation

This topic contains instructions for installing ASG-JOB/SCAN Plug-in in IBM Explorer for z/OS environment. These instructions include product requirements, installing the product, and verifying the installation.

## Installation Requirements

Before you install JOB/SCAN Plug-in, verify that the following requirements are met:

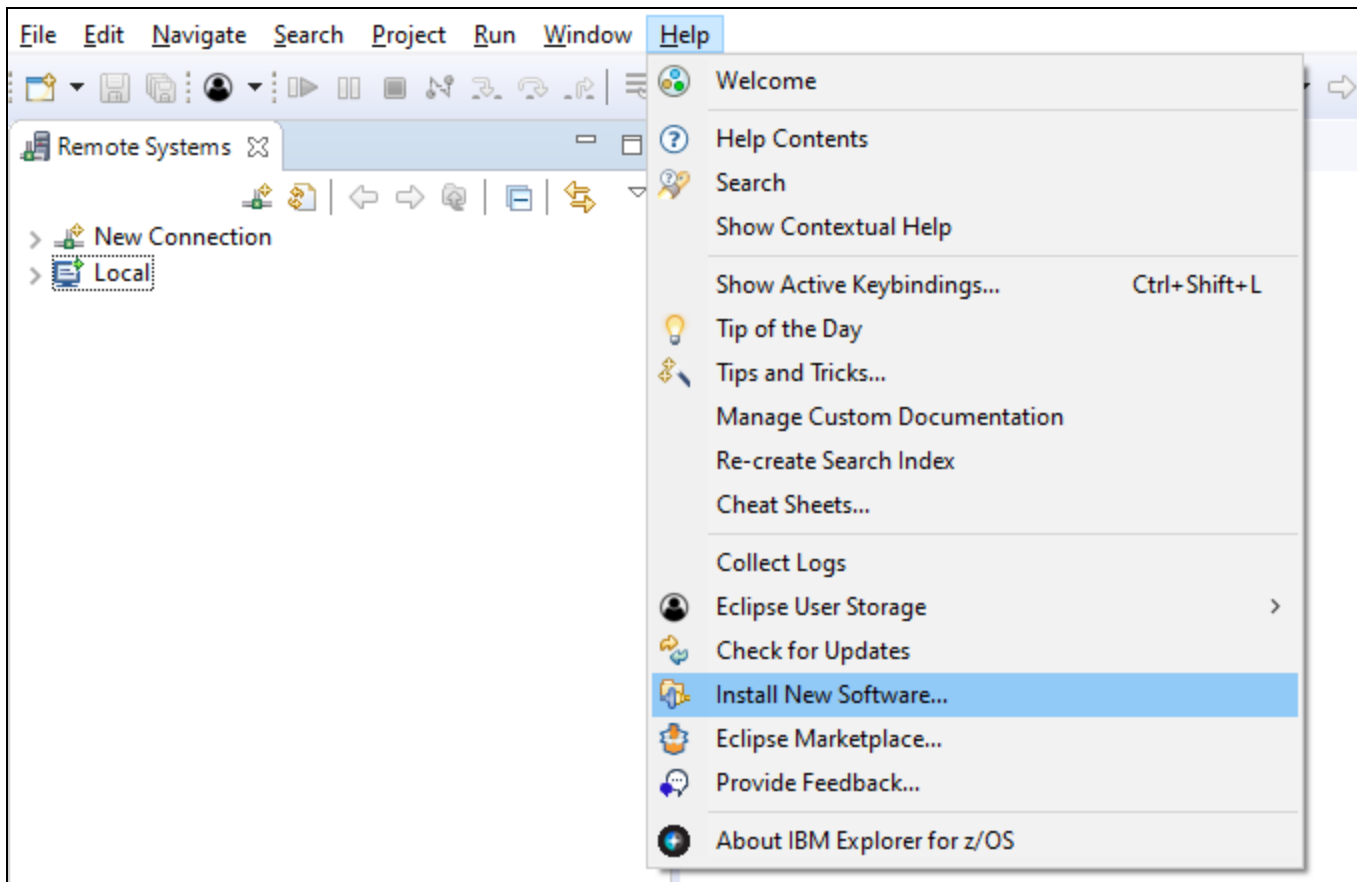
- ASG-JOB/SCAN Version 8.0.0 must be installed in the z/OS system. This topic does not explain the installation procedure for ASG-JOB/SCAN. See [ASG-JOB/SCAN Installation](#) for information on installing ASG-JOB/SCAN.
- Latest monthly cumulative service for ASG-JOB/SCAN must be applied. Visit ACCESS Portal: <https://access.asg.com> to verify any monthly maintenance apply to this product release.
- Install and Configure IBM Liberty Profile Server to enable ASG-JOB/SCAN REST Services. For details on installing Liberty server see the [Installer Reference Guide](#) section for guides on installing the ASG-JOB/SCAN REST Services. For more information about Liberty Servers z/OS installation requirements see [IBM Installing Liberty](#).
- User must have a TSO user ID defined to the mainframe system and should be able to access z/OS Unix System Services with home directory setup.
- User's TSO ID must be able to access and use ASG-JOB/SCAN under ISPF.
- IBM Explorer for z/OS 3.2 version must be installed.
- Java Runtime Environment (JRE) must be installed.
  - You can install JRE from OpenJ9: <https://www.eclipse.org/openj9/>
  - You can install JRE from Oracle: <https://www.java.com/en/download/manual.jsp>
- IBM z/OS® Management Facility (z/OSMF) Version 2.4 must be installed and running before using ASG-JOB/SCAN Plug-in. User must have read and write access to z/OSMF. To obtain read or write access, contact your system administrator. To verify if z/OSMF is configured as per ASG-JOB/SCAN Plug-in requirements, follow the procedure in [Verifying the z/OSMF Configuration](#).

## Installing ASG-JOB/SCAN Plug-in

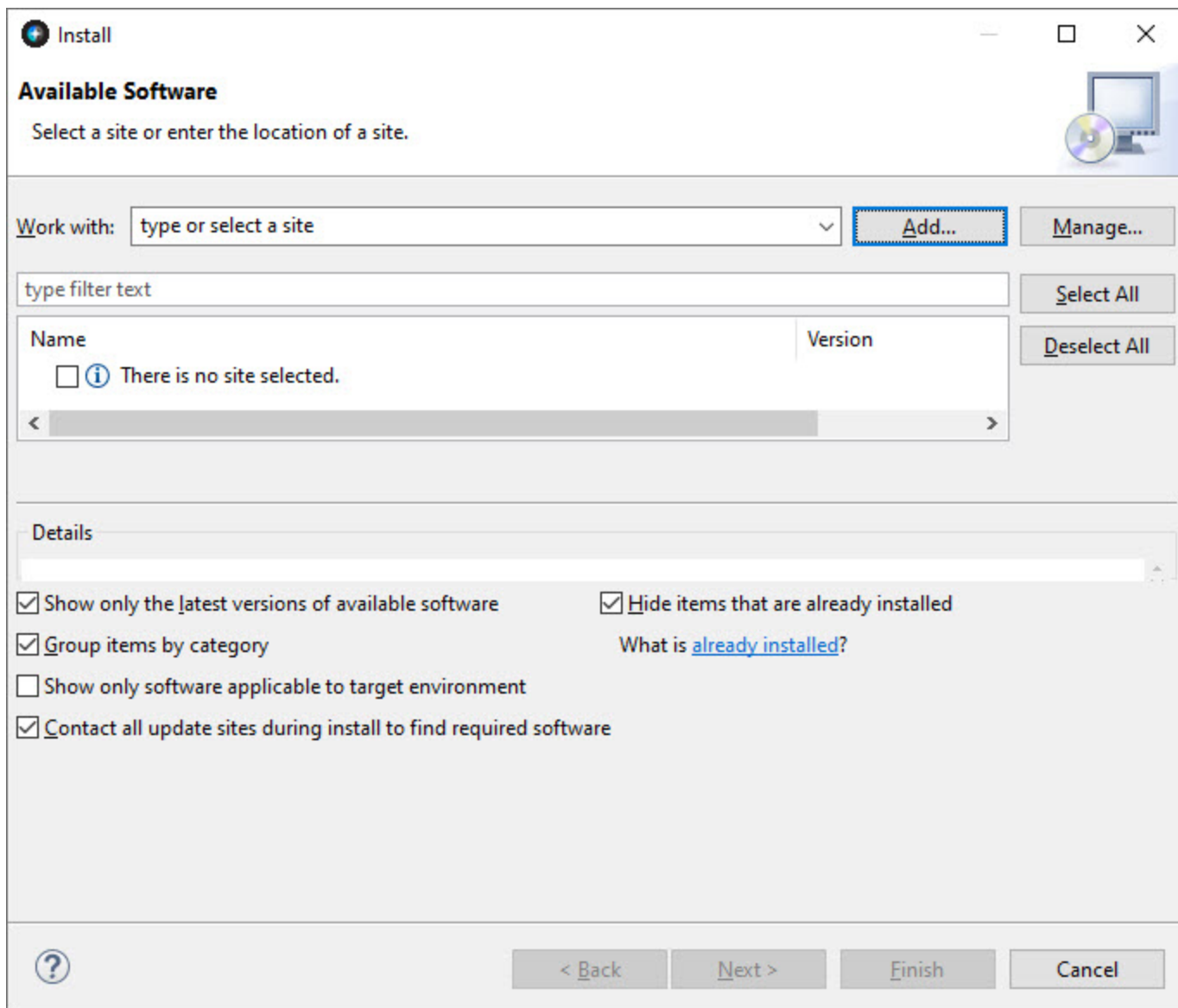
Follow these instructions to install the ASG-JOB/SCAN Plug-in after you download the product files.

### *To install ASG-JOB/SCAN Plug-in:*

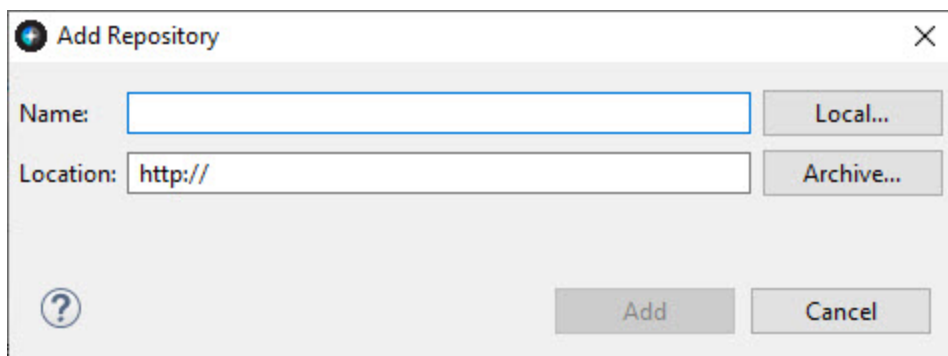
1. Start IBM Explorer for z/OS.
2. From the menu, select **Help > Install New Software....**



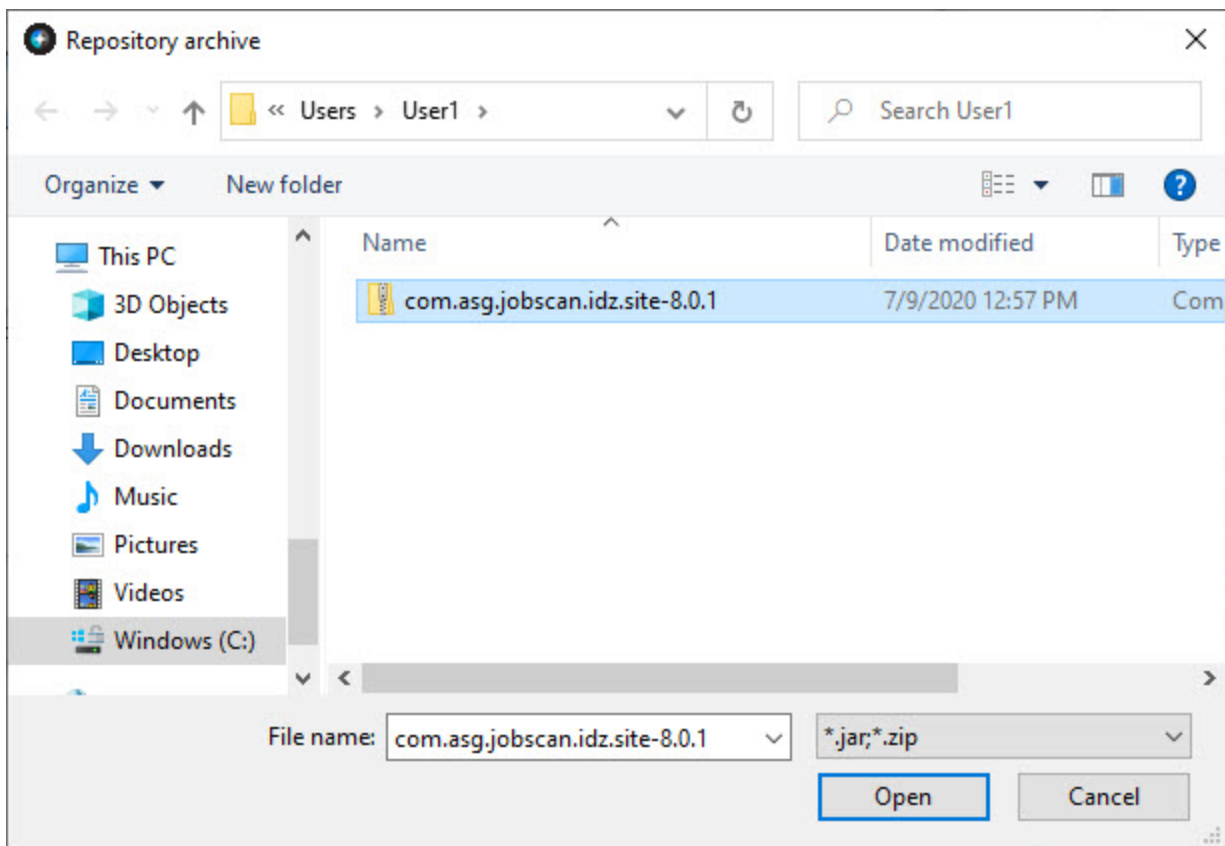
3. In the **Install** dialog, click **Add...**



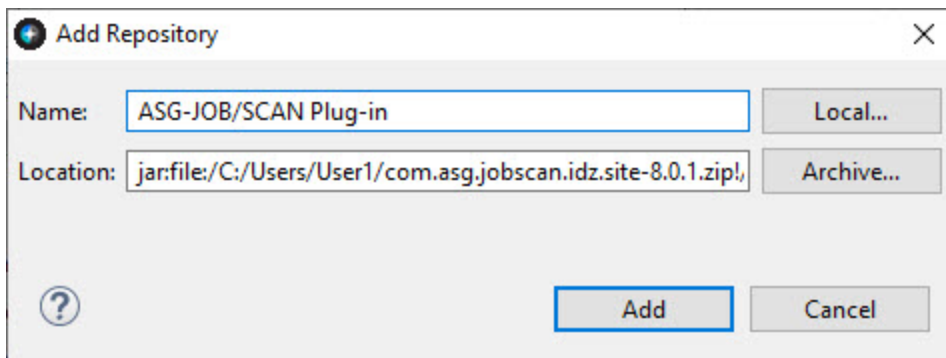
4. In the **Add Repository** dialog, enter a **Name** for the repository and click **Archive...**



5. In the **Repository archive** dialog, browse to select the ZIP file containing the install files, and click **Open**.

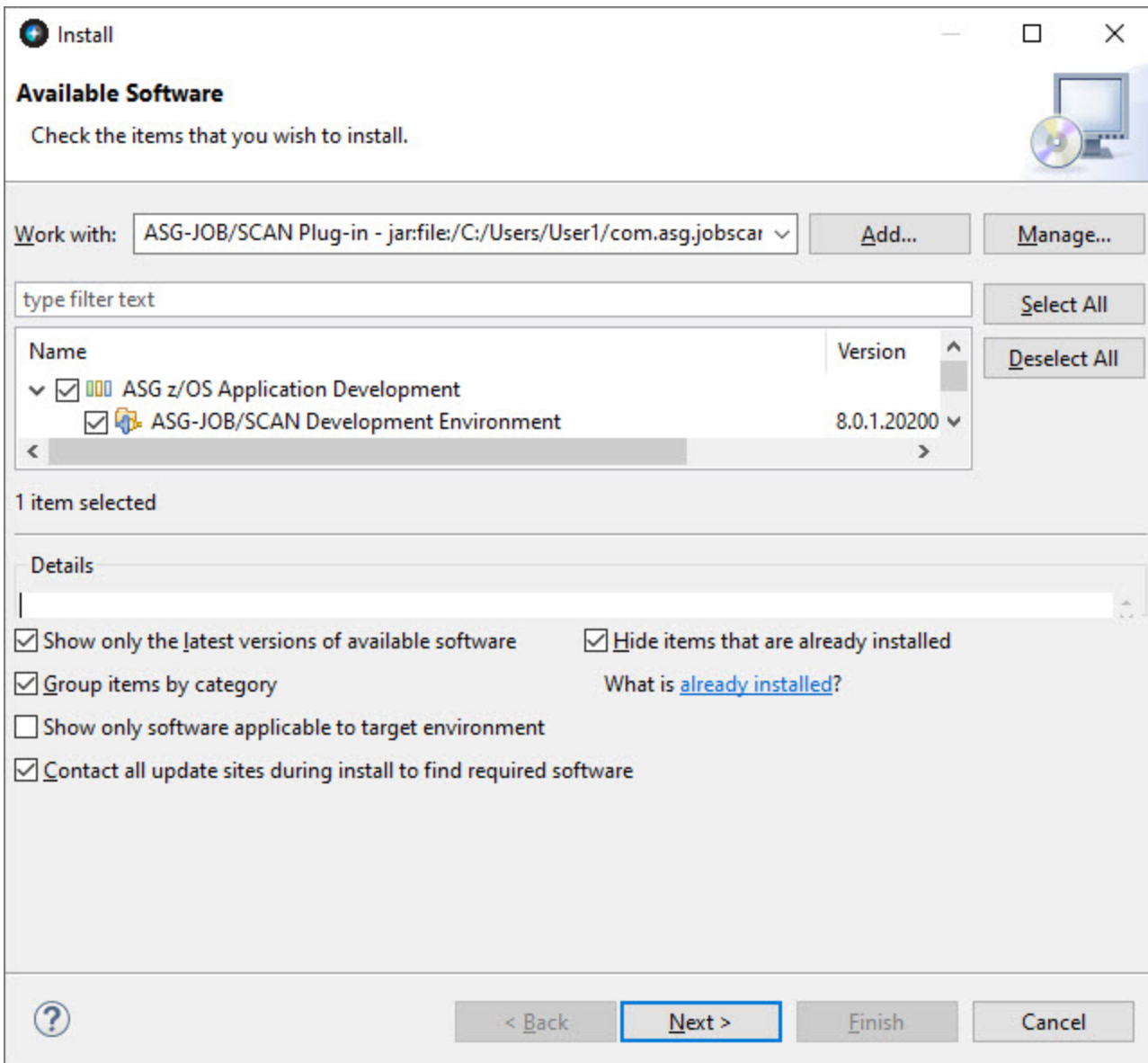


6. In the **Add Repository** dialog, click **Add**.

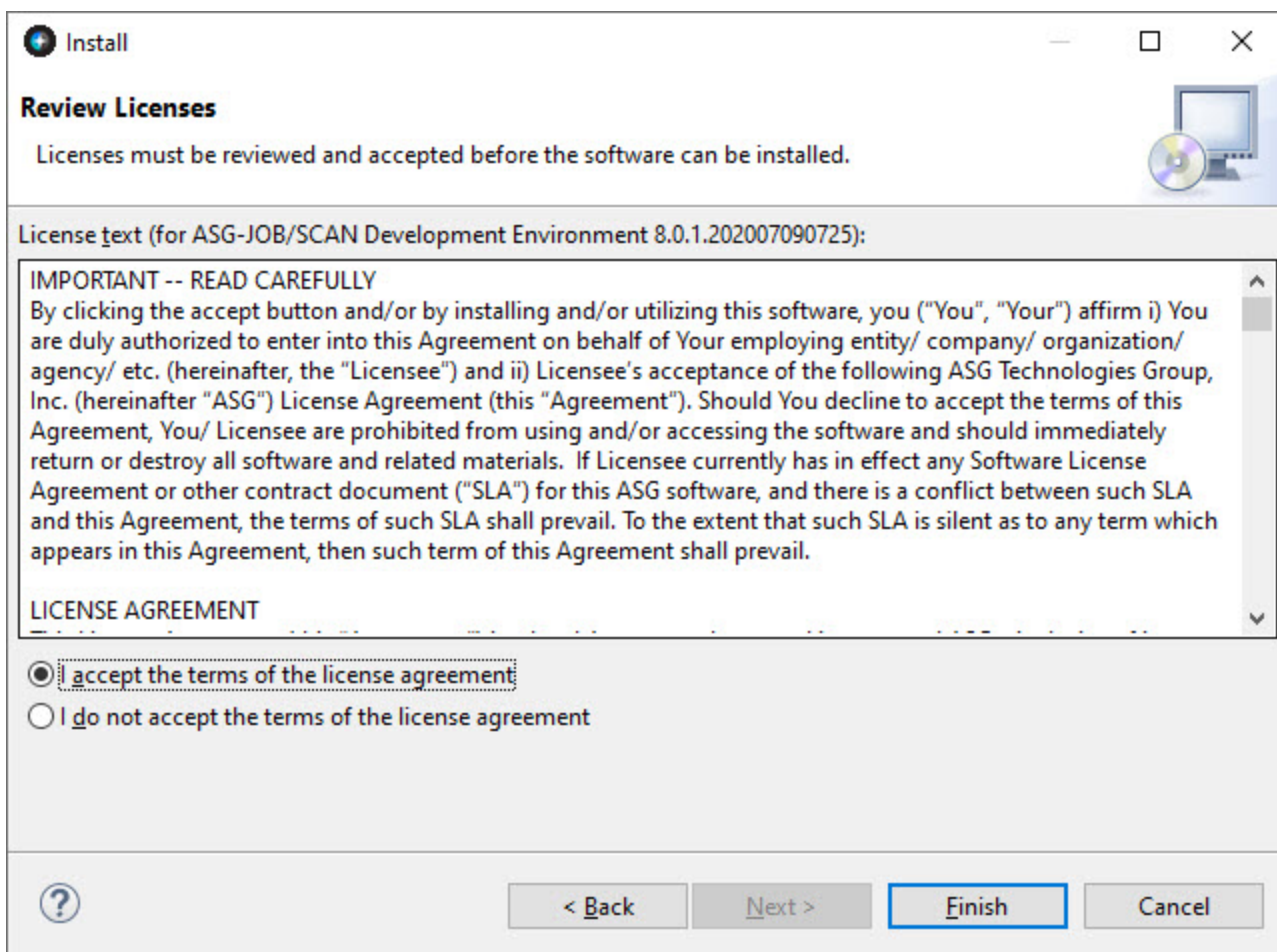


The **ASG z/OS Application Development** category appears in the **Name** area.

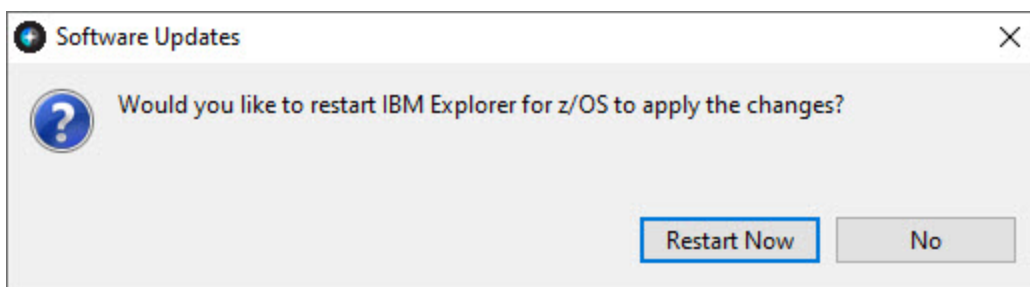
7. Select the **ASG z/OS Application Development** category and make sure that **ASG-JOB/SCAN Development Environment** feature is selected.
8. Click **Next**.



9. Select **ASG z/OS Application Development**, review the **Details** and click **Next**.
10. Review the license and select **I accept the terms of license agreement**.



11. Click **Finish**. You may receive a security warning, click **Install anyway** to continue with the installation.
12. When the installation is complete, you are prompted to restart the z/OS Explorer. Click **Restart Now**.



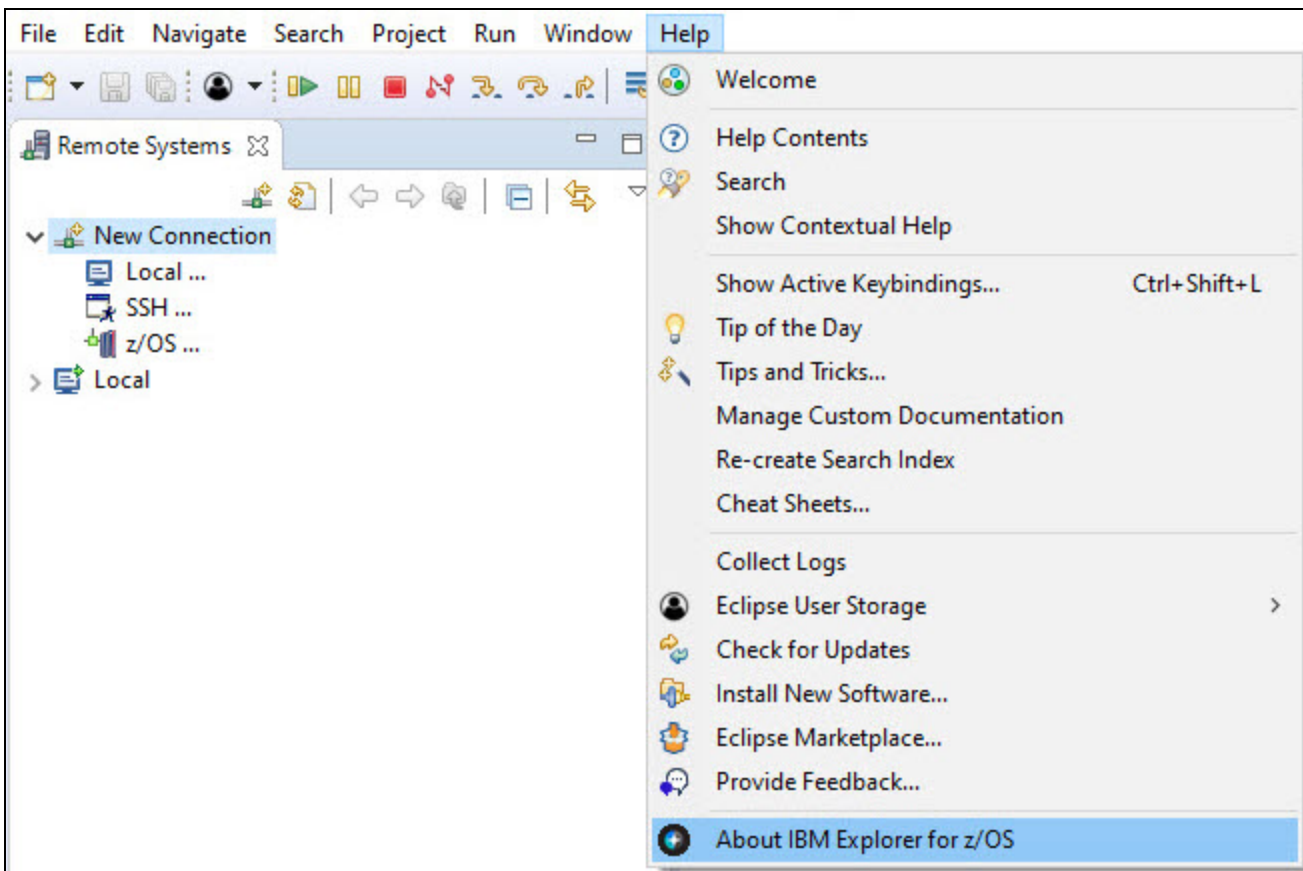
## Verifying the Installation

Follow the instructions in this section to verify whether the ASG-JOB/SCAN Plug-in is installed successfully or to find whether the plug-in is already installed.



**To verify that the JOB/SCAN Plug-in is installed:**

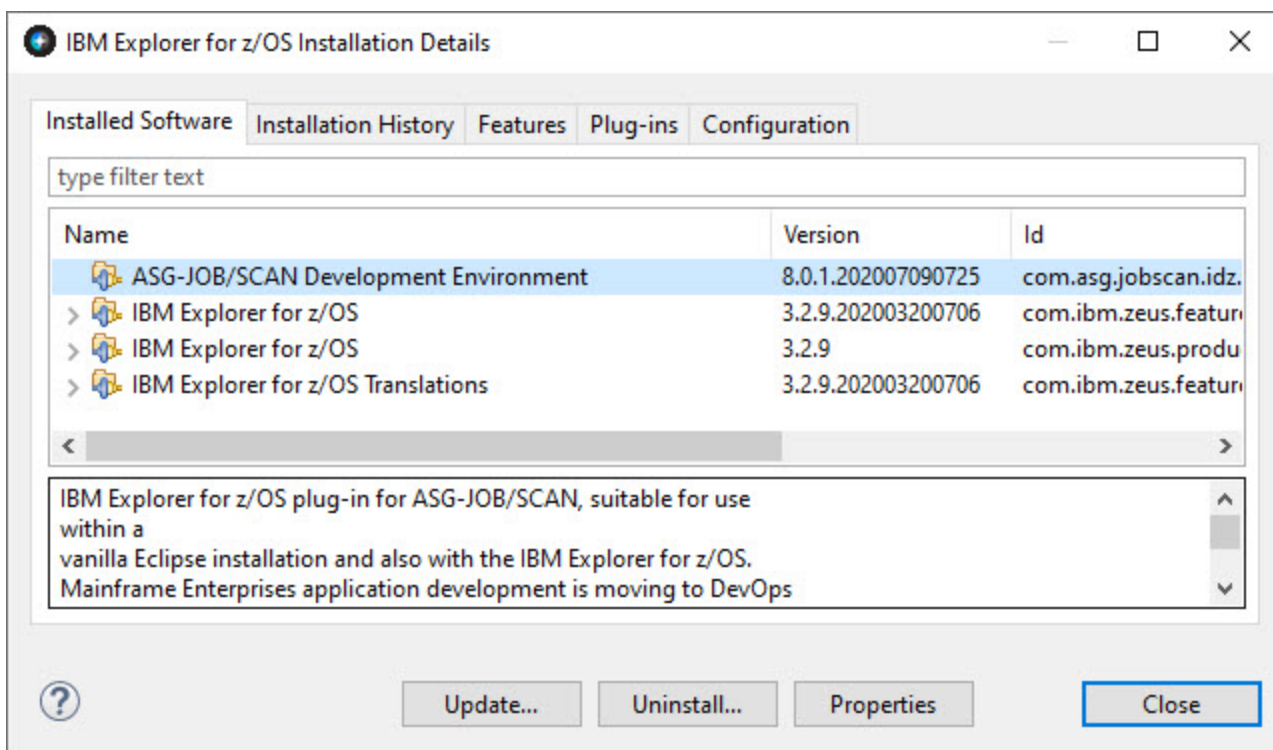
1. Start z/OS Explorer.
2. From the menu, select **Help > About IBM Explorer for z/OS**.



3. In the **About IBM Explorer for z/OS** dialog, click **Installation Details**.



- In the **Installed Software** tab, you should see **ASG-JOB/SCAN Development Environment** in the list of installed software. This indicates that JOB/SCAN Plug-in has been installed.





You can check the **Installation History** tab (Help > About IBM Explorer for z/OS > Installation Details > Installation History) for information on when a feature is installed.

## Verifying the z/OSMF Configuration

IBM z/OS® Management Facility (z/OSMF) must be installed and configured before using ASG-JOB/SCAN Plug-in. This topic provides instructions to verify if z/OSMF is configured as per ASG-JOB/SCAN Plug-in requirements.



This section does not explain the z/OSMF configuration procedure. See [Configuring z/OSMF](#) in IBM Knowledge Center for more information.

### Step to verify z/OSMF Configuration

- Verify if z/OSMF server is running

Issue the console command `/D A, IZU*` in the command line and check if the started tasks jobs, **IZUANG1** and **IZUSVR1**, are running. If the jobs are not active, issue `/S IZUANG1` and `/S IZUSVR1` commands to activate.

- Verify if z/OSMF REST services are enabled

Open a web browser to z/OS system (host name and port) and add the following REST call to the URL: `GET /zosmf/restjobs/jobs`. You should see the list of the jobs that are submitted from your user ID.

- Verify if z/OSMF data set and file REST services are enabled

See if the job **IZURFSEC** (available in the sample jobs package) is submitted and completes with the return code '0000'. To verify if services are running, issue a REST service request. See [List the z/OS data sets](#) in IBM Knowledge Center for syntax on how to issue a request.

The job IZUSVR1 shows the actual REST services that are running in z/OSMF. JOB/SCAN Plug-in requires both restjobs and restfiles services. To make sure that these services are available, check the output (STDOUT) files of IZUSRV1 job, and see if it contains a message that looks like:

```
CWWKT0016I: Web application available (default_host): https://<your-zosmf-
url>:<zosmf-port>/zosmf/<service>/
```

where <service> is a REST service name.

For more assistance work with your administrator.

- Verify access to z/OSMF

1. In a browser, open the URL: `https://<hostname>:<port>/zosmf/`

Where:

hostname is the hostname or IP address of the system in which z/OSMF is installed

port is the secure port for the z/OSMF configuration

2. Enter the z/OS USER ID that was used to configure z/OSMF
3. Enter the z/OS PASSWORD associated with the z/OS user ID.
4. Click LOGIN.

The Welcome page of IBM z/OS Management Facility is displayed. This confirms that you are authenticated to access z/OSMF.

## References:

- What setup is needed for z/OSMF - [https://www.ibm.com/support/knowledgecenter/SSLTBW\\_2.3.0/com.ibm.zos.v2r3.izua300/IZUHPINFO\\_WhatSetupsNeeded.htm](https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.izua300/IZUHPINFO_WhatSetupsNeeded.htm)
- Software prerequisites for z/OSMF - [https://www.ibm.com/support/knowledgecenter/en/SSLTBW\\_2.2.0/com.ibm.zos.v2r2.izua300/IZUHPINFO\\_SoftwarePrereqs.htm](https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.2.0/com.ibm.zos.v2r2.izua300/IZUHPINFO_SoftwarePrereqs.htm)
- Security configuration requirements for z/OSMF - [https://www.ibm.com/support/knowledgecenter/SSLTBW\\_2.3.0/com.ibm.zos.v2r3.izua300/izuconfig\\_SecurityStructuresForZosmf.htm](https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.izua300/izuconfig_SecurityStructuresForZosmf.htm)
- Configuring z/OSMF - [https://www.ibm.com/support/knowledgecenter/SSLTBW\\_2.3.0/com.ibm.zos.v2r3.izua300/IZUHPINFO\\_PartConfiguring.htm](https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.izua300/IZUHPINFO_PartConfiguring.htm)

## Verifying the Liberty Server Configuration

IBM Liberty Server for z/OS must be installed and a Liberty server for ASG-JOB/SCAN REST Services must be configured before using ASG-JOB/SCAN Plug-in. This topic provides instructions to verify if the Liberty server for ASG-JOB/SCAN REST Services is ready as per the ASG-JOB/SCAN Plug-in requirements.



This section does not explain the Liberty Server for ASG-JOB/SCAN REST Services configuration procedure. See [Installer Reference Guide](#) for more information.

A Liberty server instance may be started as a z/OS started task or started from the z/OS UNIX shell. To operate Liberty servers as z/OS started tasks, BBGZSRV by default would be the started task name.

When a Liberty server seeks to use z/OS authorized services, BBGZANGL by default is required. ASG-JOB/SCAN REST Services does not require a Liberty angel process to be setup.



For more information on the various options for configuring and starting a Liberty runtime environment on the z/OS, see the [IBM WebSphere Application Server for z/OS Liberty](#).

## Step to Verify Liberty Server for ASG-JOB/SCAN REST Services

- Verify if Liberty server is running

Issue the console command `/D A, BBGZSRV*` in the command line and check if one or more started tasks **BBGZSRV\*** are running. If you are not using **BBGZSRV\*** sample proc, substitute this name with the name of the “JCL Start proc name” dedicated to ASG-JOB/SCAN REST Services.

- Start a Liberty server

If started task **BBGZSRV\*** or the name of the “JCL Start proc name” dedicated to ASG-JOB/SCAN REST Services is not running you need to start it. The following example shows the command that you would use to start a Liberty server configured for ASG-JOB/SCAN REST Services named “jjsrest” using the sample `bbgzsrv` proc:

```
START BBGZSRV.JJSREST,PARMS='jjsrest'
```

The following example shows the command that you would use to stop a ASG-JOB/SCAN REST Server (note: z/OS START command syntax of “S membername[.identifier]”, was used to identify `jjsrest` as the name identifying the task to be started, so you can issues commands against this task)

```
P JJSREST
```

- Review the messages of the Liberty server. If ASG-JOB/SCAN REST Services were successfully started these messages would be logged to the output STDOUT DD.

The following message verifies that JOB/SCAN REST Services is up and running:

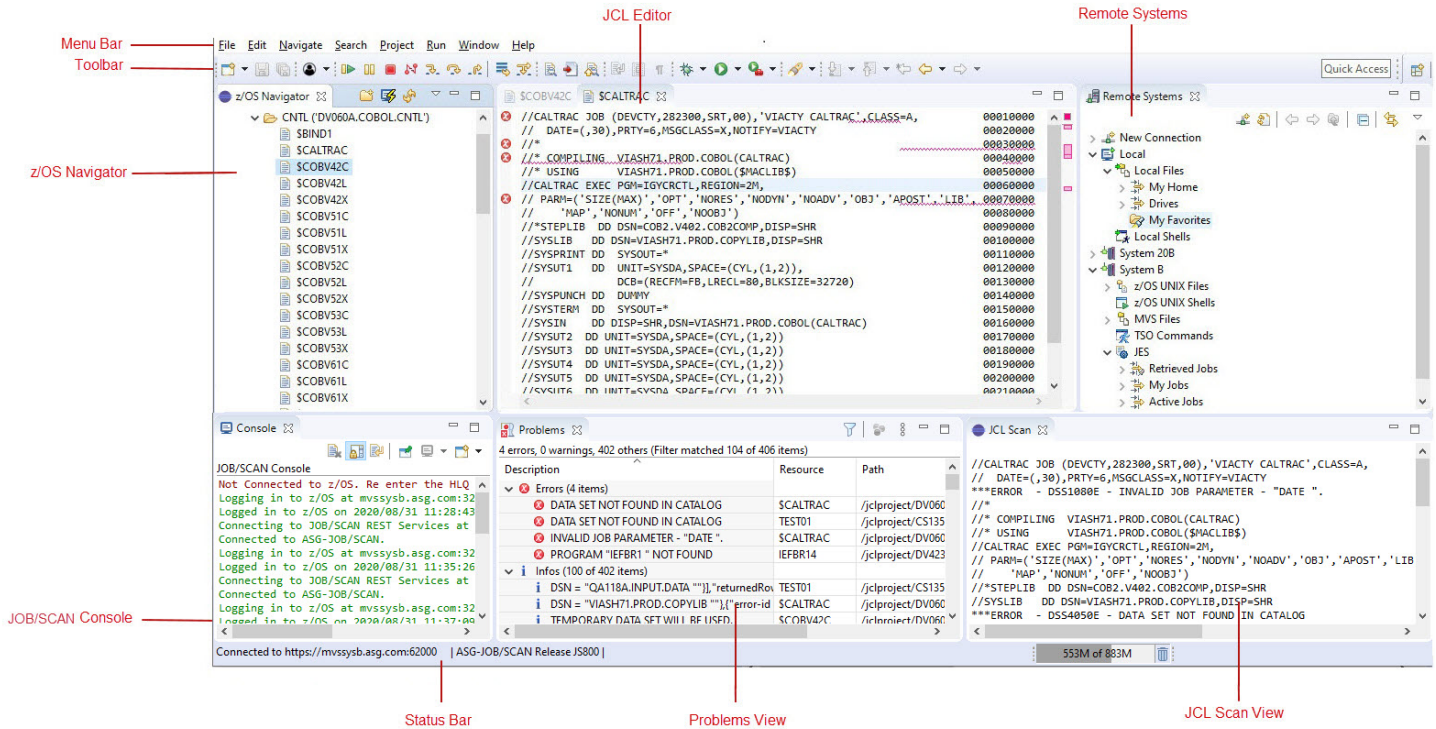
```
CWWKF0011I: The jjsrest server is ready to run a smarter planet.
```

#### Sample STDOUT DD Output

```
Launching jjsrest(WAS FOR Z/OS 19.0.0.12/wlp-1.0.35.cl191220191120-0300) on IBM J9 VM, version
8.0.6.0 - pmz6480sr6-20191107_01(SR6) (en_US)
[AUDIT ] CWWKE0001I: The server jjsrest has been launched.
[AUDIT ] CWWKZ0058I: Monitoring dropins for applications.
[AUDIT ] CWWKT0016I: Web application available (default_host):
http://mvssyse.asg.com:49080/api/docs/
[AUDIT ] CWWKT0016I: Web application available (default_host):
http://mvssyse.asg.com:49080/api/explorer/
[AUDIT ] CWWKT0016I: Web application available (default_host):
http://mvssyse.asg.com:49080/jobscan/
[AUDIT ] CWWKZ0001I: Application jobscan started in 0.556 seconds.
[AUDIT ] CWWKF0015I: The server has the following interim fixes active in the runtime:
PH19989,PH19528. For a full listing of installed fixes run: productInfo version --ifixes
[AUDIT ] CWWKF0012I: The server installed the following features: [distributedMap-1.0, el-3.0,
jaxrs-2.1, jaxrsClient-2.1, jndi-1.0, jsonp-1.1, jsp-2.3, openapi-3.0, servlet-4.0, ssl-1.0,
zosSecurity-1.0].
[AUDIT ] CWWKF0011I: The jjsrest server is ready to run a smarter planet. The jjsrest server
started in 3.358 seconds.
[AUDIT ] CWPKI0820A: The default keystore has been created using the 'keystore_password'
environment variable.
```

# User Interface

This topic describes about the commonly used interface elements that are required when using the JOB/SCAN Plug-in functions. It also provides information on how to view these elements when not displayed in the workbench.



## Menu Bar

The **Menu Bar** contains menus and menu options that are used to perform a task. JOB/SCAN commands are displayed as additional options in the Edit menu. You must be connected to JOB/SCAN's server before running the JOB/SCAN commands. Some options in the menu bar are available only when a file is active in the JCL Editor.




## Toolbar

The **Toolbar** provides quick access to tasks that are commonly performed by a user. JOB/SCAN commands are displayed as buttons on the toolbar. You must be connected to JOB/SCAN's server before running the JOB/SCAN commands. Some options on the toolbar are available only when a file is active in the JCL Editor.

## z/OS Navigator View

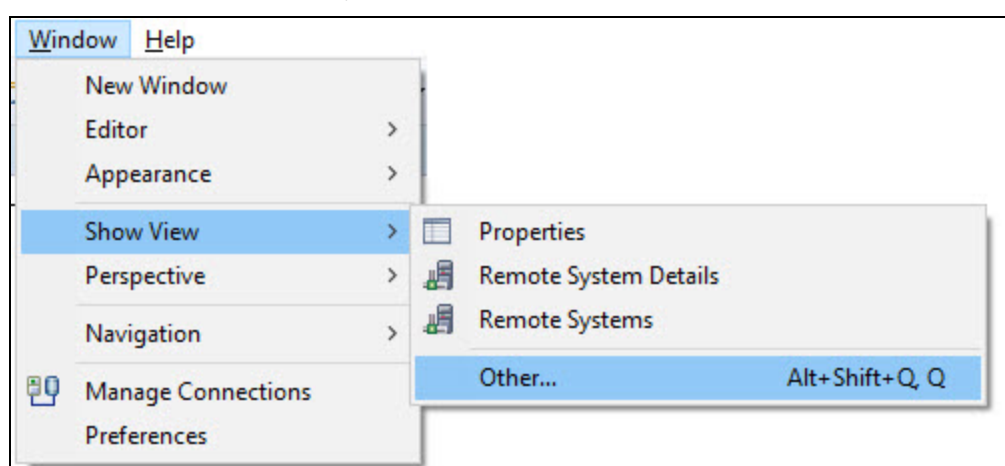
**z/OS Navigator** view displays datasets that are fetched from z/OS. The datasets are displayed under a High-Level Qualifier (HLQ) in a tree-like structure. For more information about displaying datasets, refer to [Displaying Datasets](#).

z/OS Navigator View consists of the following icons:

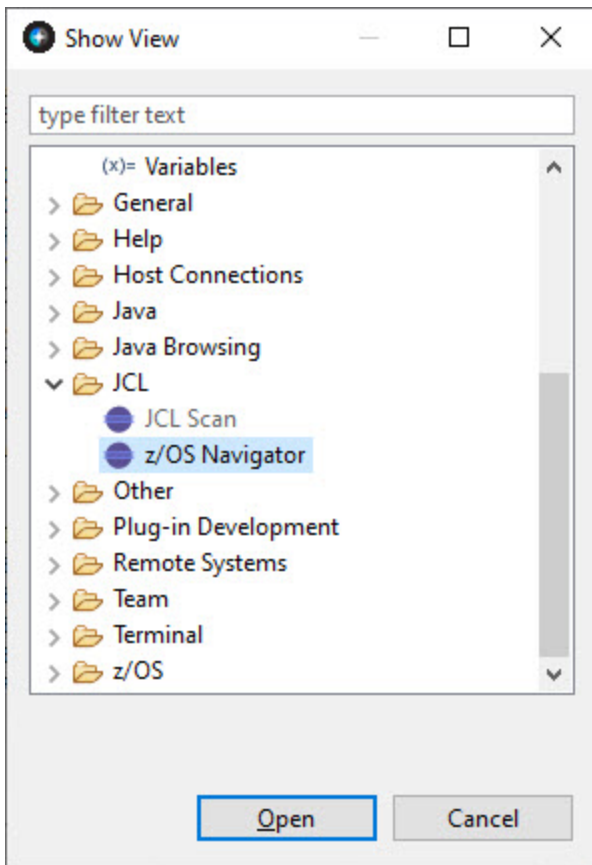
Icon	Name	Description
	Add High Level Qualifier	Fetches datasets from the z/OS system and displays under the specified HLQ. See <a href="#">Working with z/OS Datasets</a> .
	Connect to z/OS	Establishes connection to z/OS system. See <a href="#">Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator</a> .
	Refresh View	Refreshes the list of datasets displayed. See <a href="#">Refreshing Dataset</a> .

### To display z/OS Navigator view:

1. From the menu, select **Window > Show View > Other...**



2. In the **Show View** dialog, do one of the following:
  - Enter **z/OS Navigator** in the **type filter text** box.
  - Locate and expand the **JCL** folder.



3. Double-click on **z/OS Navigator** or select **z/OS Navigator** and click **Open**.

## Remote Systems

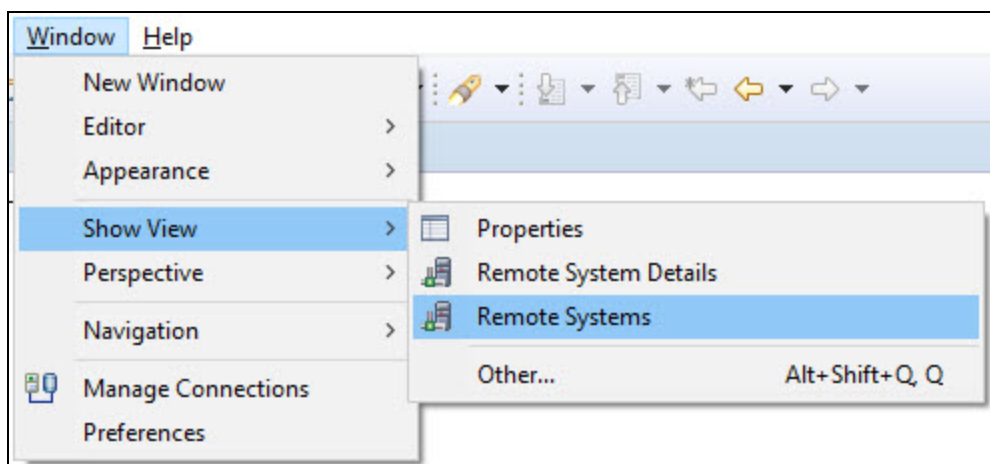
**Remote Systems** view displays datasets that are in z/OS system and files that are in your local file system.

For more information about displaying z/OS datasets, refer to [Displaying Datasets](#). For information about displaying local files, refer to [Displaying Local Files](#).

**To display Remote Systems view:**

- From the menu, select **Window > Show View > Remote Systems**.





## Status Bar

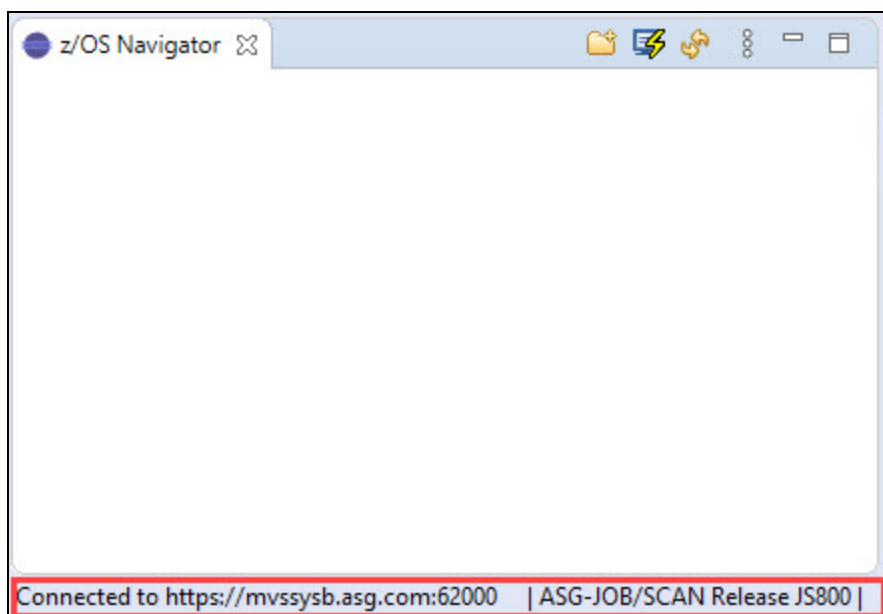
**Status Bar** displays the z/OS connection status.



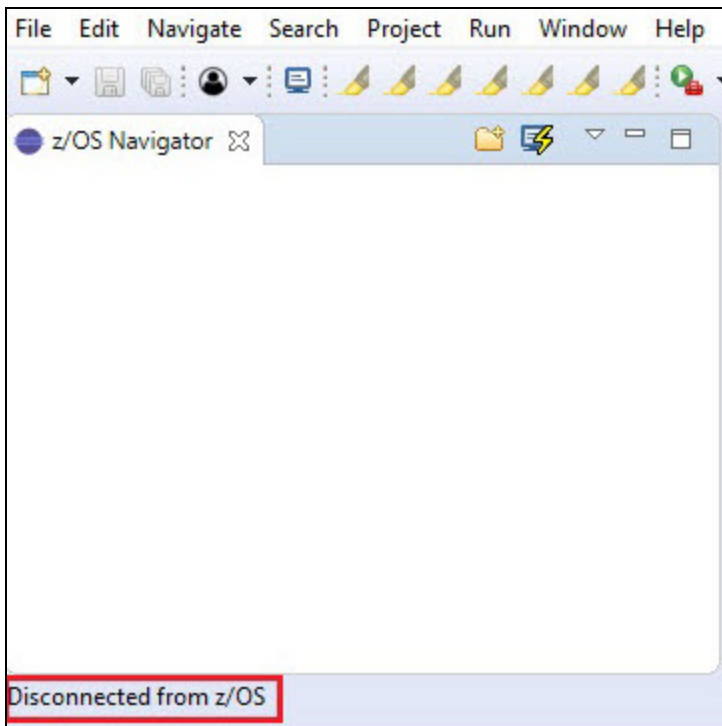
The z/OS connection status is shown only when the **z/OS Navigator** view is active.

After establishing the connection to z/OS and JOB/SCAN server, the status bar displays the following message:

**Connected to** <host address or IP address of the z/OS system>:<port number of the host machine on which the JOB/SCAN REST API service is running> | **ASG-JOB/SCAN Release JS**<version of the ASG-JOB/SCAN it is connected> |



When the connection is lost or when the session is disconnected from z/OS, the status bar display the message: **Disconnected from z/OS**.



## JCL Editor

**JCL Editor** is typically used to view, edit, save and validate a JCL file. You can view more than one file in the Editor area, but only one can be active at a time. Tabs in the editor area indicate the names of files that are currently open for editing. Some options in the menu bar and toolbar are available only when a file is active in the JCL Editor.

File Edit Navigate Search Project Macro Tools Run Configure Compuware Topaz Connect Window Help

z/OS Navigator

SCALTRAC \*SCOBV42C

```

//COBV42C JOB (DEVCTY,282300,SRT,00),'COBOL 4.2 COMP',CLASS=A,PRTY=6, 00010000
// NOTIFY=&SYSUID,MSGCLASS=X,REGION=0M,TIME=20 00020000
//* 00030000
//COBCOMP PROC SRC='DV060A.NTIS', 00040000
// HLQ='DV060A.COBV42', 00050000
// MEM= 00060000
//STEPCOMP EXEC PGM=IGYCRCTL,REGION=0M, 00070000
// PARM='NOOPT,LIST,MAP,NUM,NOOFF,S' 00080000
//* PARM='LIST,MAP,S,NOADV,FASTSRT,RENT,TRUNC(OPT),NOXREF' 00090000
//*TEPLIB DD DSN=COB2.R32.COB2COMP,DISP=SHR 00100000
//*TEPLIB DD DSN=SYSP.IGY122.SIGYCOMP,DISP=SHR 00110000
//*TEPLIB DD DSN=ESW.MVSCOB.IGY210.SIGYCOMP,DISP=SHR 00120000
//*TEPLIB DD DSN=SYSP.IGY331.SIGYCOMP,DISP=SHR 00130000
//*TEPLIB DD DSN=SYSP.IGY340.SIGYCOMP,DISP=SHR 00140000
//*TEPLIB DD DSN=SYSP.IGY410.SIGYCOMP,DISP=SHR 00150000
//STEPLIB DD DSN=SYSP.IGY420.SIGYCOMP,DISP=SHR,DCB=BLKSIZE=32760 00160000
// DSN=CEE.SCEERUN,DISP=SHR 00170000
// DD DSN=CEE.SCEERUN2,DISP=SHR 00180000
//SYSIN DD DSN=&SRC..COBOL(&MEM),DISP=SHR 00190000
//SYSLIB DD DSN=&SRC..COBOL,DISP=SHR 00200000
//*YSPRINT DD SYSOUT=* 00210000
//*YSPRINT DD DSN=&HLQ..LIST(&MEM),DISP=SHR 00220000
//SYSPRINT DD DSN=&LSTOUT,UNIT=SYSALLDA, 00230000
// DISP=(MOD,PASS),SPACE=(CYL,(1,1)) 00240000
//SYSLIN DD DSN=&HLQ..OBJ(&MEM),DISP=SHR 00250000
//SYSUT1 DD UNIT=SYSALLDA,SPACE=(CYL,(2,2)) 00260000
//SYSUT2 DD UNIT=SYSALLDA,SPACE=(CYL,(2,2)) 00270000
//SYSUT3 DD UNIT=SYSALLDA,SPACE=(CYL,(2,2)) 00280000
//SYSUT4 DD UNIT=SYSALLDA,SPACE=(CYL,(2,2)) 00290000
//SYSUT5 DD UNIT=SYSALLDA,SPACE=(CYL,(2,2)) 00300000
//SYSUT6 DD UNIT=SYSALLDA,SPACE=(CYL,(2,2)) 00310000
//SYSUT7 DD UNIT=SYSALLDA,SPACE=(CYL,(2,2)) 00320000
//SYSUT8 DD UNIT=SYSALLDA,SPACE=(CYL,(2,2)) 00330000
//SYSUT9 DD UNIT=SYSALLDA,SPACE=(CYL,(2,2)) 00340000
//SYSUT10 DD UNIT=SYSALLDA,SPACE=(CYL,(2,2)) 00350000
//SYSUT11 DD UNIT=SYSALLDA,SPACE=(CYL,(2,2)) 00360000
//SYSUT12 DD UNIT=SYSALLDA,SPACE=(CYL,(2,2)) 00370000
//SYSUT13 DD UNIT=SYSALLDA,SPACE=(CYL,(2,2)) 00380000

```

## Problems View

**Problems** view displays the list of problem messages that are identified in a JCL file. This view is automatically displayed when you scan a file. For more information about Problems view, refer to [Reports](#).

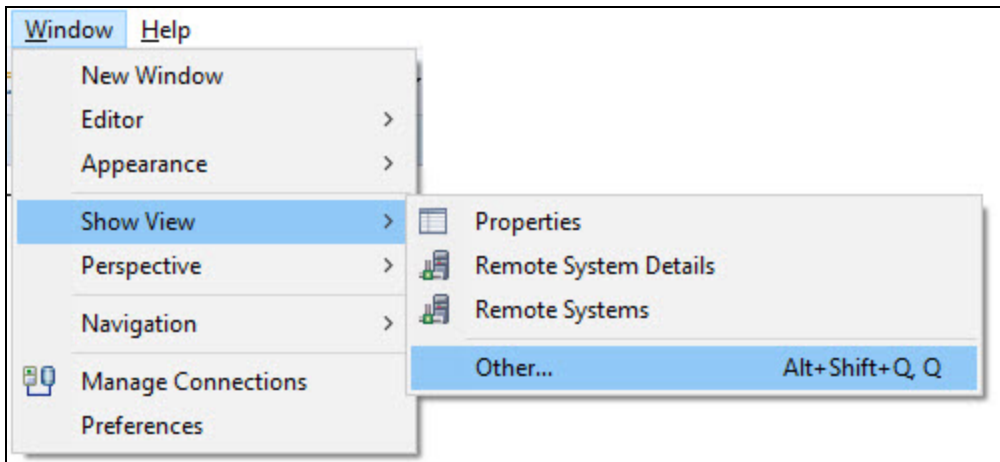
Problems

2 errors, 0 warnings, 401 others (Filter matched 102 of 403 items)

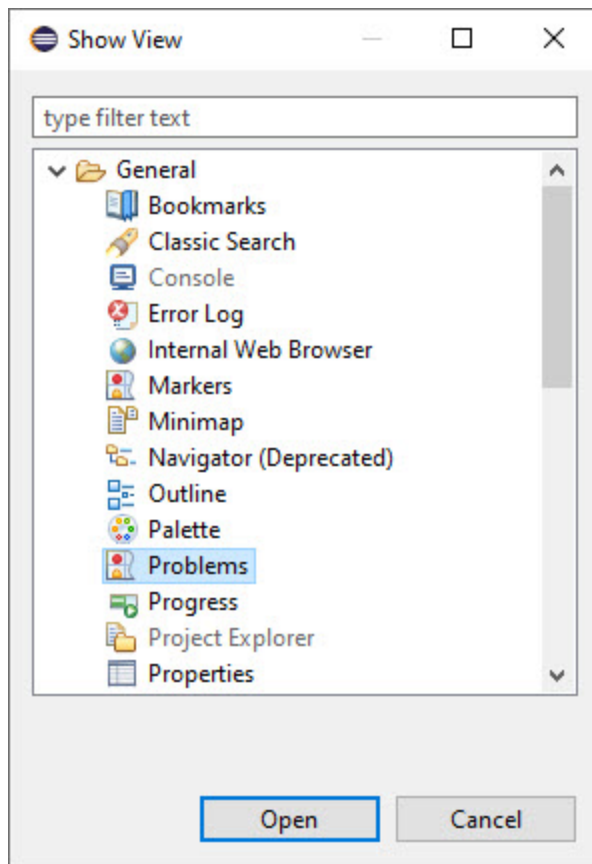
Description	Resource	Path	Location	Type
Errors (2 items)				
DATA SET NOT FOUND IN CATALOG	SCALTRAC	/jclproject/DV060A....	line 10	JCL Scan Mes...
INVALID JOB PARAMETER - "DATE ".	SCALTRAC	/jclproject/DV060A....	line 2	JCL Scan Mes...
Infos (100 of 401 items)				
DSN = "VIASH71.PROD.COPYLIB ""},{ "error-id	SCALTRAC	/jclproject/DV060A....	line 10	JCL Scan Mes...
TEMPORARY DATA SET WILL BE USED.	SCOBV42C	/jclproject/DV060A....	line 65	JCL Scan Mes...
TEMPORARY DATA SET WILL BE USED.	SCOBV42C	/jclproject/DV060A....	line 65	JCL Scan Mes...
TEMPORARY DATA SET WILL BE USED.	SCOBV42C	/jclproject/DV060A....	line 65	JCL Scan Mes...
TEMPORARY DATA SET WILL BE USED.	SCOBV42C	/jclproject/DV060A....	line 65	JCL Scan Mes...

**To display Problems view:**

1. From the menu, select **Window > Show View > Other....**



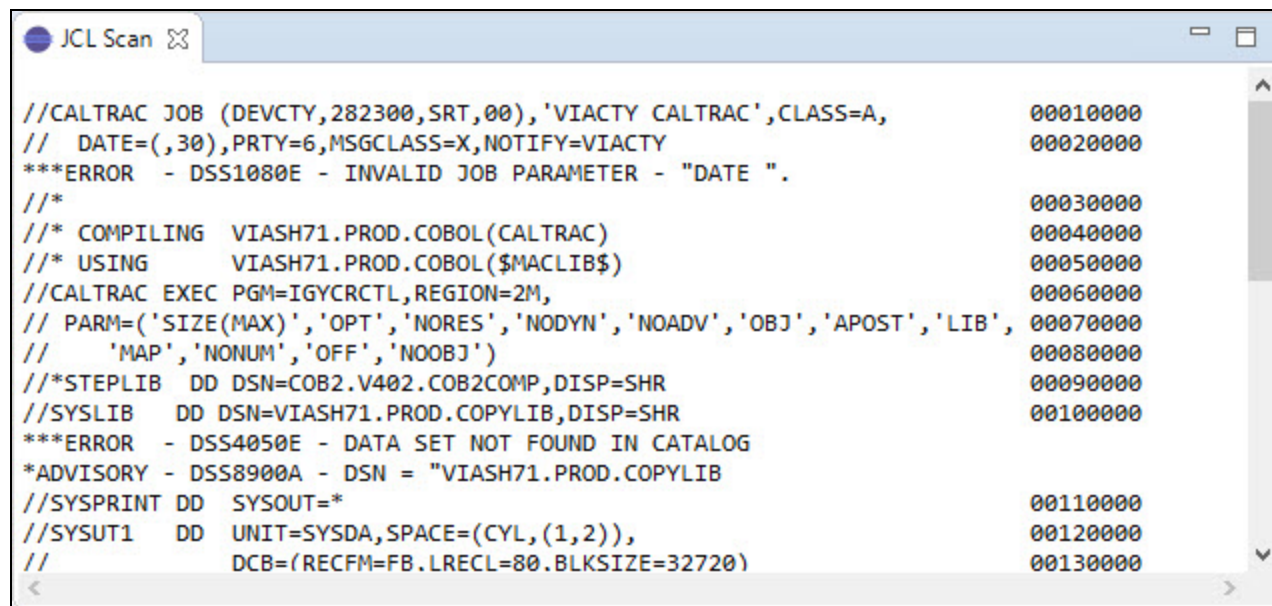
2. In the **Show View** dialog, do one of the following:
  - Enter **Problems** in the **type filter text** box.
  - Locate and expand the **General** folder.



3. Double-click on **Problems** or select **Problems** and click **Open**.

## JCL Scan View

**JCL Scan** view displays the scanned report of a JCL file. This report is similar to the JOB/SCAN's Structured JCL Report (SJL). For more information about JCL Scan view, refer to [Reports](#).



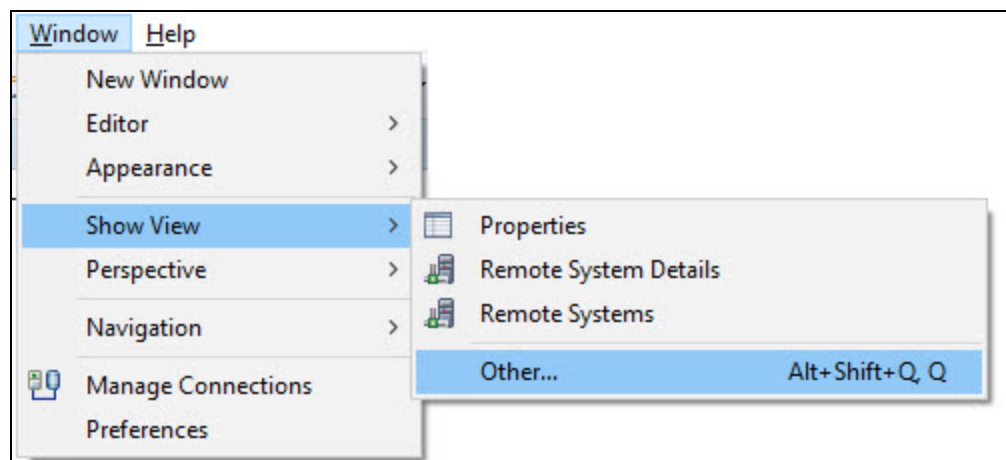
```

JCL Scan
//CALTRAC JOB (DEVCTY,282300,SRT,00), 'VIACTY CALTRAC', CLASS=A,          00010000
// DATE=(,30),PRTY=6,MSGCLASS=X,NOTIFY=VIACTY                          00020000
***ERROR - DSS1080E - INVALID JOB PARAMETER - "DATE ".
//*                                                                       00030000
//* COMPILING VIASH71.PROD.COBOL(CALTRAC)                               00040000
//* USING      VIASH71.PROD.COBOL($MACLIB$)                             00050000
//CALTRAC EXEC PGM=IGYCRCTL,REGION=2M,                                  00060000
// PARM=('SIZE(MAX)', 'OPT', 'NORES', 'NODYN', 'NOADV', 'OBJ', 'APOST', 'LIB', 00070000
//   'MAP', 'NONUM', 'OFF', 'NOOBJ')                                    00080000
//*STEPLIB DD DSN=COB2.V402.COB2COMP,DISP=SHR                           00090000
//SYSLIB DD DSN=VIASH71.PROD.COPYLIB,DISP=SHR                          00100000
***ERROR - DSS4050E - DATA SET NOT FOUND IN CATALOG
*ADVISORY - DSS8900A - DSN = "VIASH71.PROD.COPYLIB
//SYSPRINT DD SYSOUT=*                                                00110000
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,2)),                               00120000
// DCB=(RECFM=FB.LRECL=80.BLKSIZE=32720)                              00130000

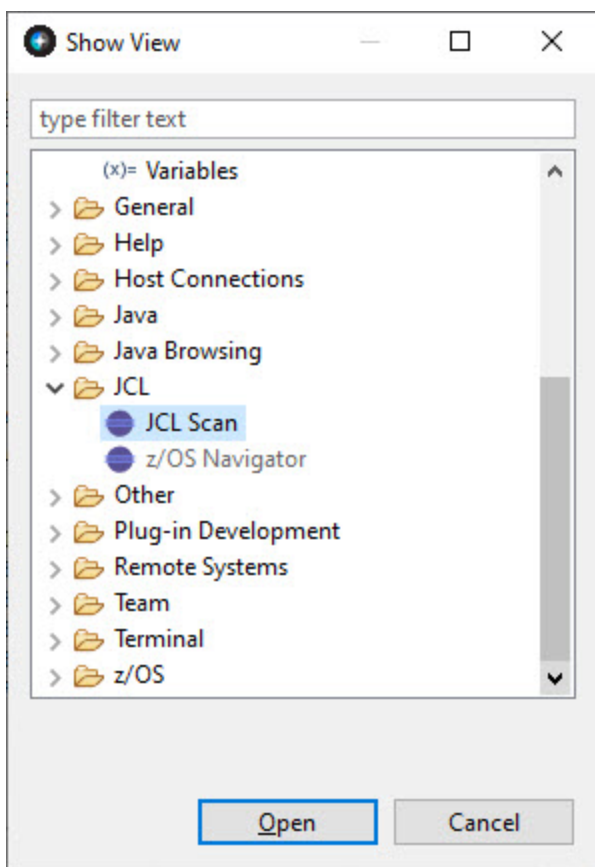
```

### To display JCL Scan view:

1. From the menu, select **Window > Show View > Other...**



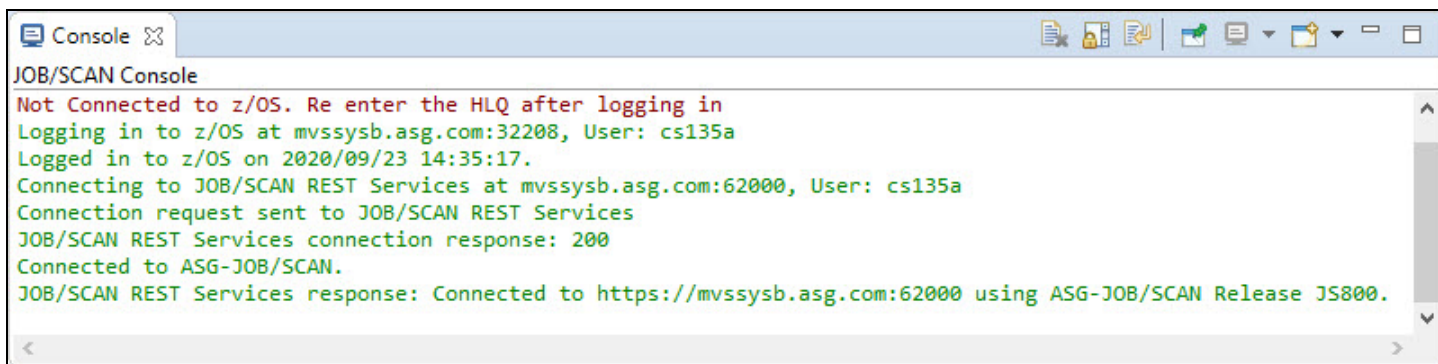
2. In the **Show View** dialog, do one of the following:
  - Enter **JCL Scan** in the **type filter text** box.
  - Locate and expand the **JCL** folder.



3. Double-click on **JCL Scan** or select **JCL Scan** and click **Open**.

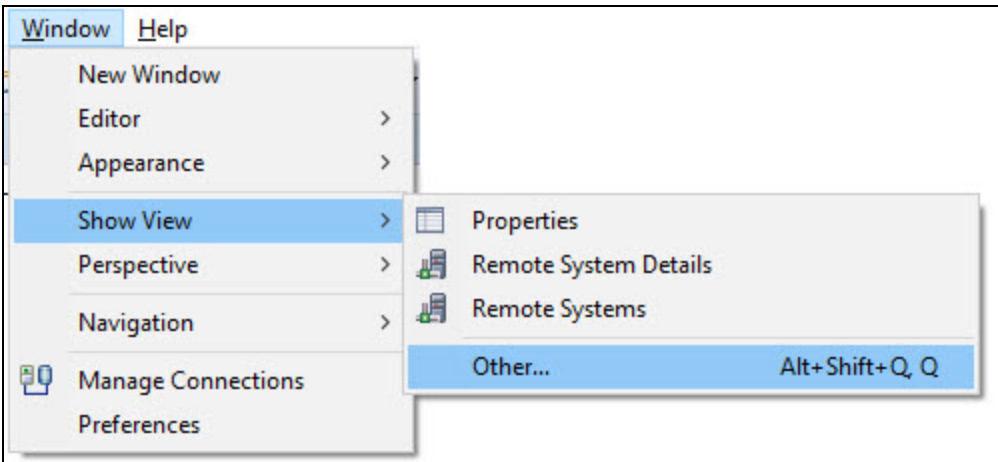
## JOB/SCAN Console View

**JOB/SCAN Console** view displays the output of a process performed in JOB/SCAN Plug-in. The console displays the output messages in two different colors: **Green** for the success messages and **Red** for the error messages.

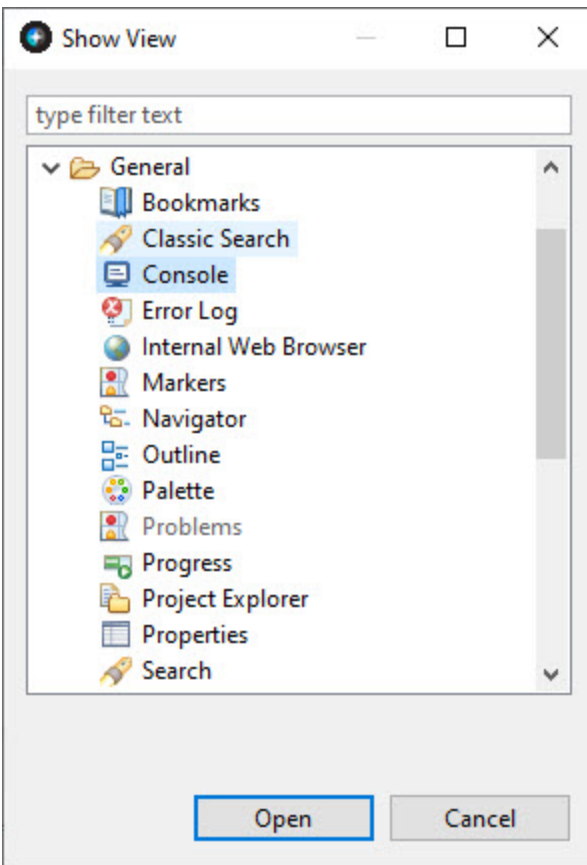



### To display JOB/SCAN Console view:

1. From the menu, select **Window > Show View > Other....**



2. In the **Show View** dialog, do one of the following:
  - Enter **Console** in the **type filter text** box.
  - Locate and expand the **General** folder.



3. Double-click on **Console** or select **Console** and click **Open**. The **Console** view pane appears.
4. In **Console** view pane, click **Open Console** icon (  ) and select **JOB/SCAN Console**.





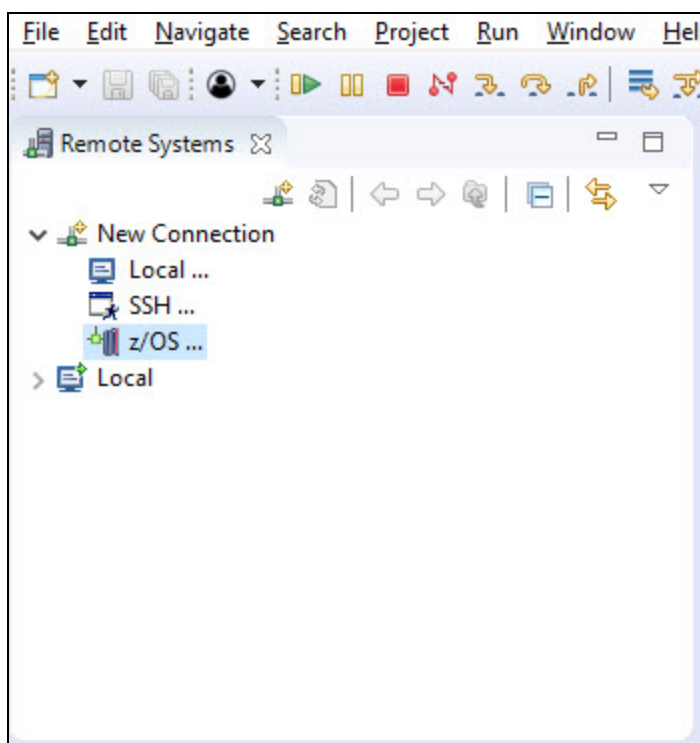
# Connecting to z/OS from Remote Systems

This topic explains the connection to z/OS system from Remote Systems view.

★ Remote Systems view which is a part of the Remote System Explorer (RSE) perspective is developed by IBM, so the sample UI presented in this guide may look different and it is only for illustrative purpose.

## To create a z/OS Connection:

1. In the **Remote Systems** tab, expand the **New Connection** node and double-click on **z/OS**. The **New Connection** dialog appears.



2. In the **Remote z/OS System Connection** properties page of **New Connection**, fill the required values and click **Next**.

**New Connection**

**Remote z/OS System Connection**

Define connection information

Parent profile: INMAASWETMUNE2P

Host name: mvssysb.asg.com

Connection name: System B

Description:

Verify host name

[Configure proxy settings](#)

Field Name	Description
Parent profile	Select a profile name from the list. If you are creating a connection for the first time, then you are prompted to create a new profile.
Host name	Enter host address or IP address of the z/OS system.
Connection name	This field is auto populated with Host name value. You can edit and enter a short name to call the system.
Description	Enter a description of the connection.
Verify host name	Select this option to verify that the host name is valid before you connect.

3. In the **Connection Configuration** properties page of **New Connection**, fill the required values and click **Finish**.

**New Connection**

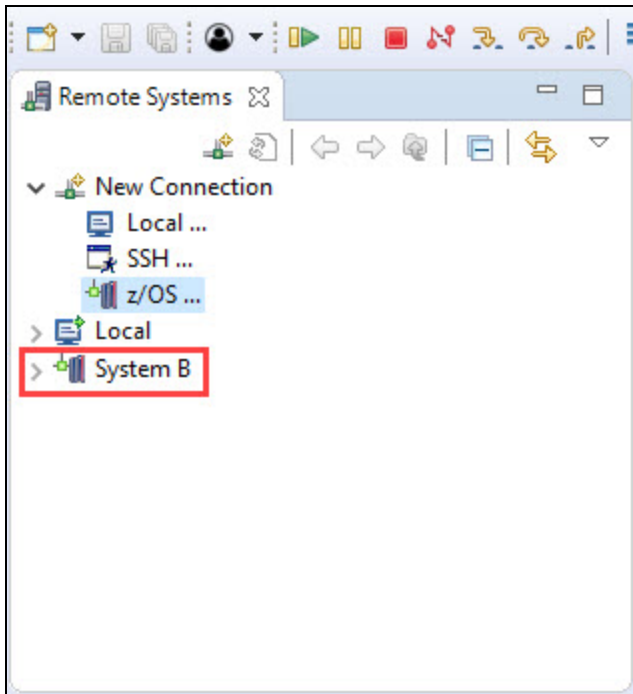
**Connection Configuration**  
Configure the connection

Indicate how the remote server should be launched

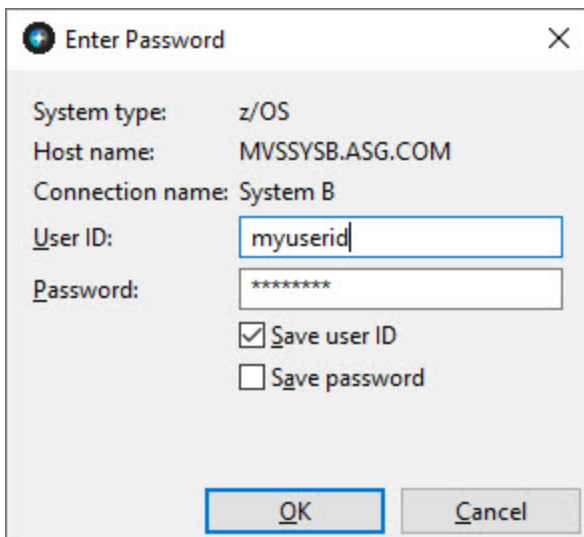
Daemon Port (1-65535)  Authentication method

Field Name	Description
Daemom Port (1-65535)	Enter the port number that the server daemon on the remote host uses.
Authentication method	Choose a method for authenticating with the remote system. Select userid/password if you log on to the remote system by using a user ID and password. Select certificate if you use client certificate authentication.

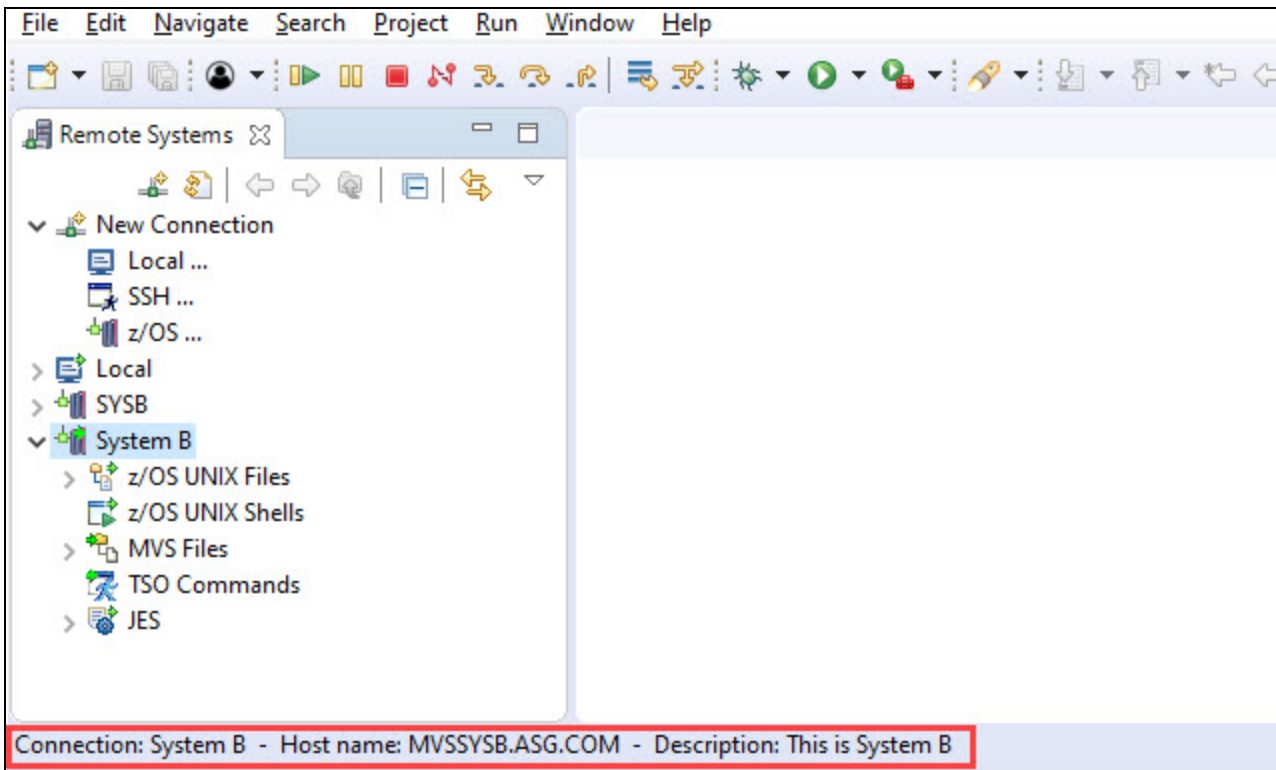
The new connection is displayed in the **Remote Systems** view.



4. Right-click on the new z/OS connection and select **Connect**. The **Enter Password** dialog appears.



5. Enter your **User ID** and **Password**.
6. Click **OK**.



If the connection is successful, then the following message is displayed at the bottom on the window:

**Connection:** <Connection name> - **Host name:** <Host name> - **Description:** <Description>


Else, a self-explanatory error message appears.

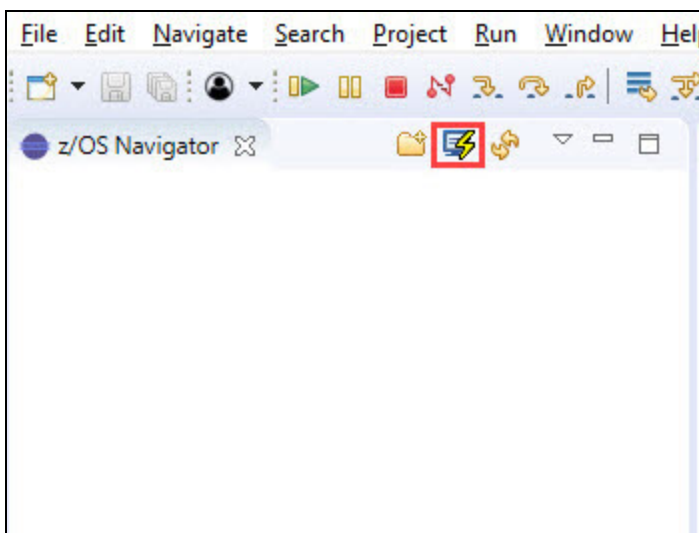
# Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator

This topic explains the connection to z/OS system and ASG-JOB/SCAN. Connection to z/OS and JOB/SCAN is essential to view z/OS resources and perform JOB/SCAN functionalities.

If you are a first time user, then you must create a connection profile that includes information about the z/OS system and ASG-JOB/SCAN.

## **To connect to z/OS and ASG-JOB/SCAN:**

1. Select  (**Connect to z/OS**) icon from the **z/OS Navigator** pane.



The **Connect to z/OS** dialog appears.

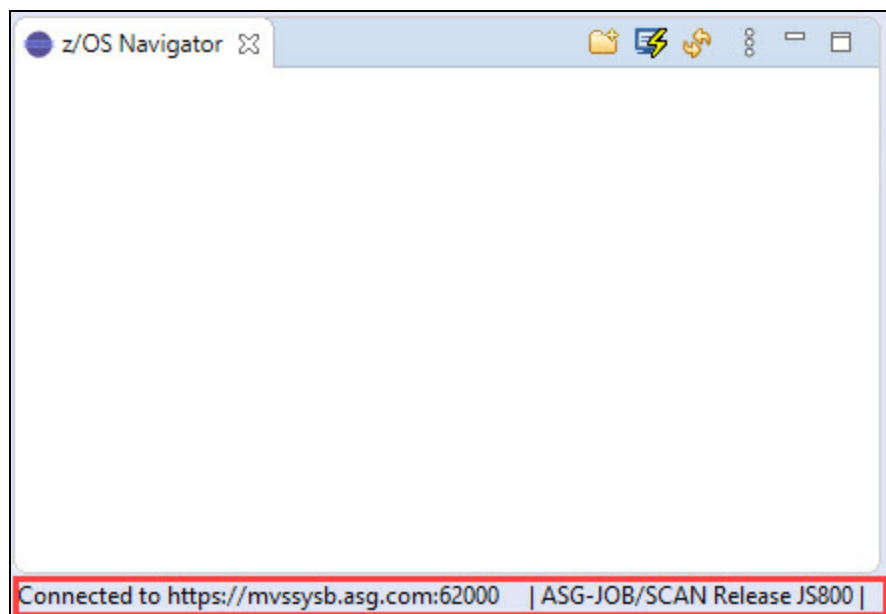
2. Select **System Name** from the list of connection profiles. **MVS User ID** is auto-populated on selecting the system name.




Initially, the **System Name** is empty. Create a new connection profile to display your profile in the **System Name** list. See the sub-topic: [To create a connection Profile](#) for more information.

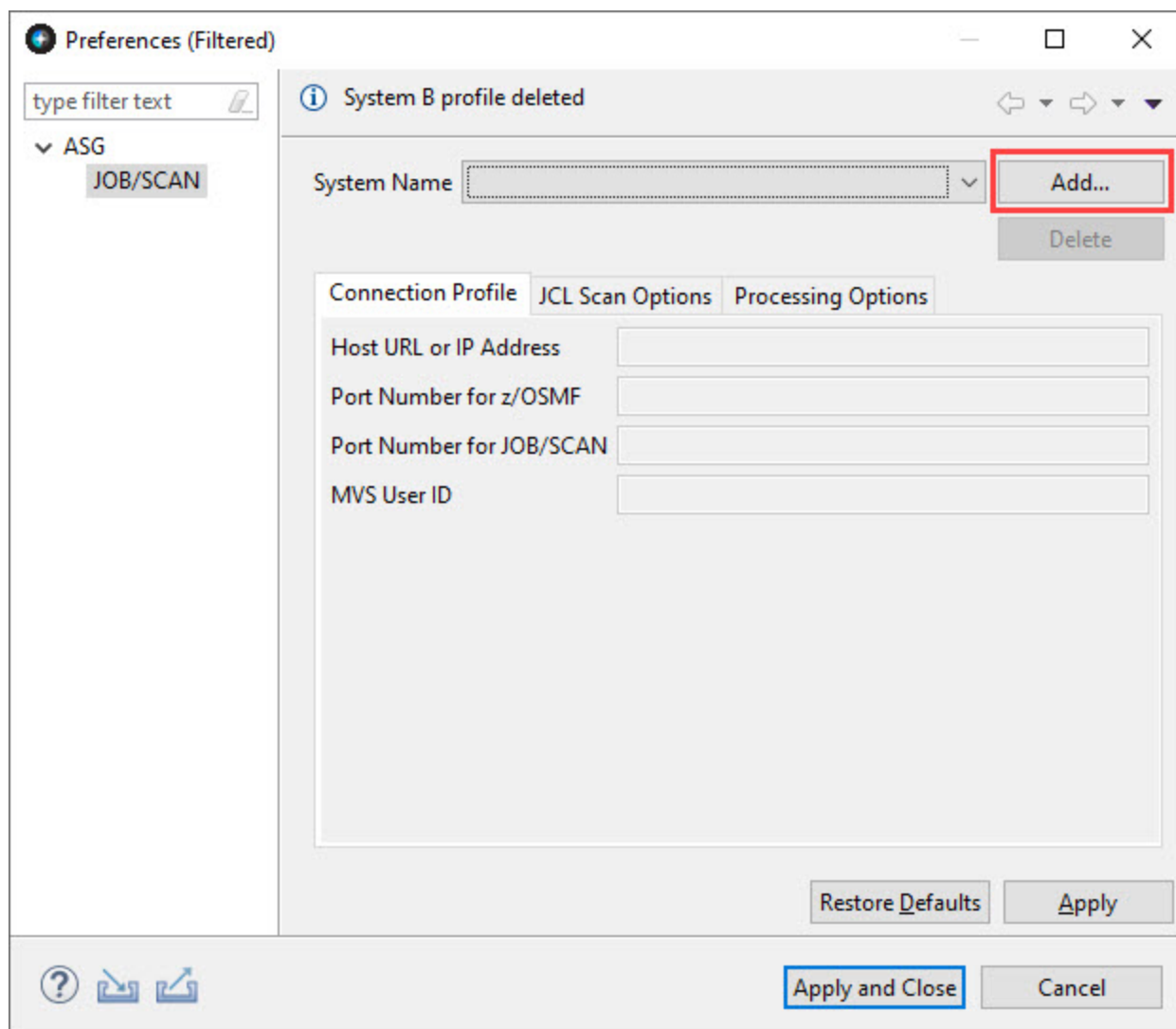
3. Enter **Password** and click **Connect**.

If logon is successful, the **Connect to z/OS** dialog is closed and the connection status along with the version of the JOB/SCAN is displayed in the status bar. Else, a self-explanatory error message appears in the **Connect to z/OS** dialog.



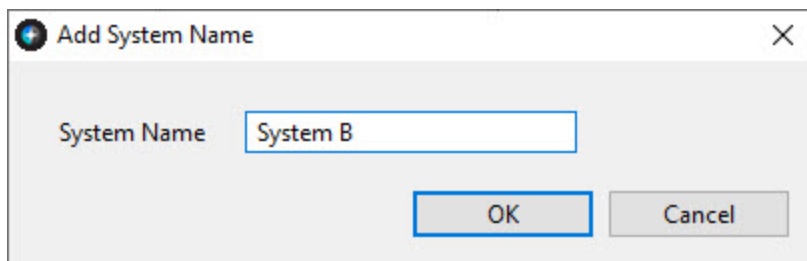
**To create a Connection Profile:**

1. Do either of the following to navigate to the **JOB/SCAN** preference page:
  - From the **z/OS Navigator** pane, select  (Connect to z/OS) icon. In the **Connect to z/OS** dialog, click **Manage Profiles**.
  - From the **Window** menu, select **Preferences**. In the **Preferences** dialog, select **ASG > JOB/SCAN**.
2. Do the following to create a new system name:
  - i. Click **Add...**. The **Add System Name** dialog appears.



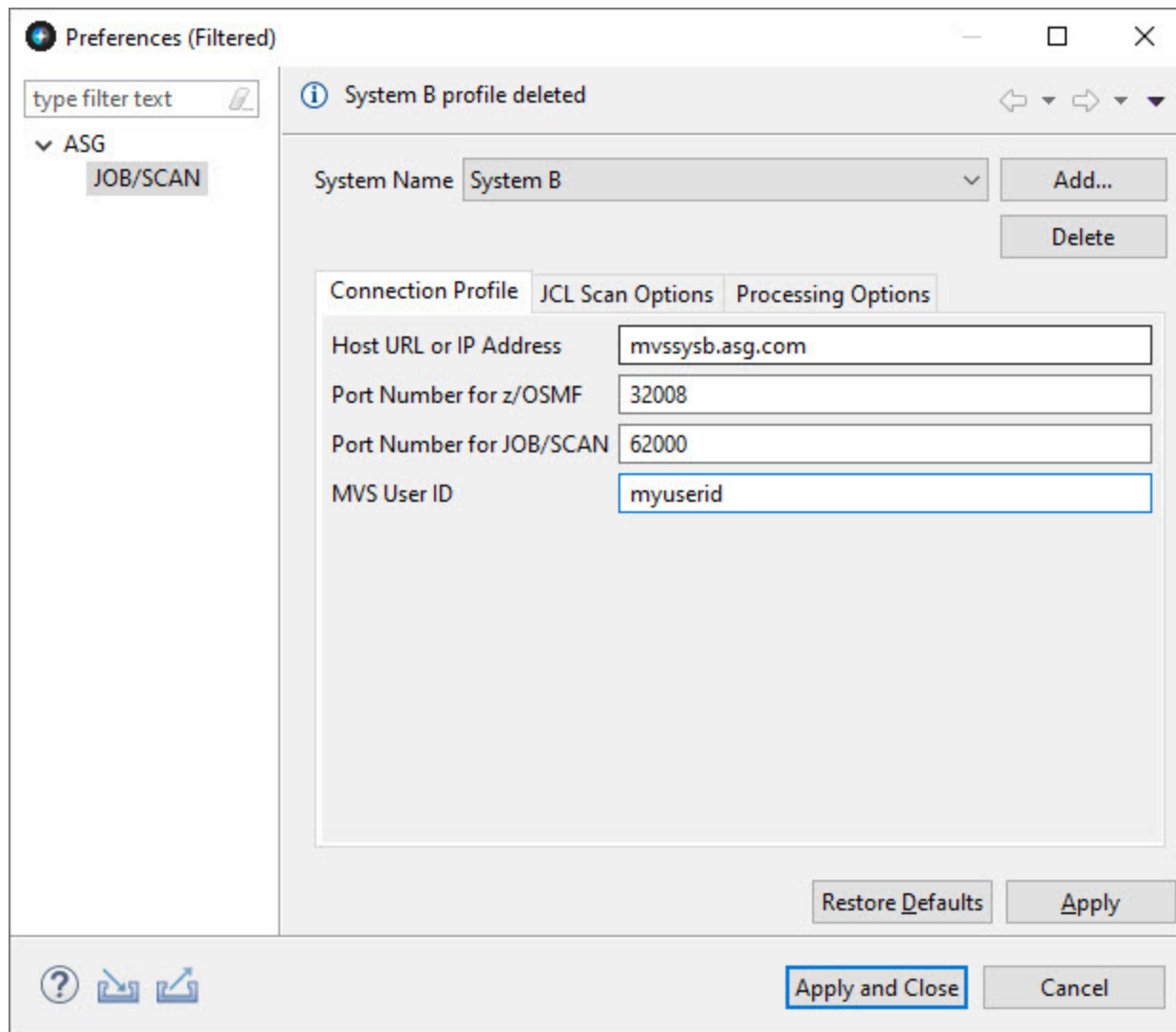
- ii. Enter a name for the new connection profile and click **OK**.





The given system name appears in the **JOB/SCAN** preference page.

3. In the **Connection Profile** tab fill the following details:



Field Name	Description
Host Name or IP Address	Enter host address or IP address of the z/OS system.

Field Name	Description
Port Number for z/OSMF	Enter port number of the host machine on which the z/OSMF REST API service is running.
Port Number for JOB/SCAN	Enter port number of the host machine on which the JOB/SCAN REST API service is running.
MVS User ID	Enter User ID for logging on to the z/OS system.

4. Click **Apply** to save the scan options or click **Apply and Close** to save the scan options and return to the **Connect to z/OS** dialog.

**To update a Connection Profile:**

1. In the **JOB/SCAN** preference page, select the **System Name** that you like to edit and update the details in the **Connection Profile** tab.
2. Click **Apply** to save the connection profile or click **Apply and Close** to save the connection profile and return to the **Connect to z/OS** dialog.



After updating the connection profile, save the changes and reconnect to ASG-JOB/SCAN using the Connect to z/OS dialog to establish connection with updated details.

**To delete a Connection Profile:**

1. In the **JOB/SCAN** preference page, select the **System Name** that you like to delete and click **Delete**.
2. In the confirmation dialog, click **OK**. Connection profile is deleted from the **System Name** list.



The connection profile details and scan options that are saved to the System Name will be lost, so make sure before you delete it.

# Setting the JCL Scan Options

This topic explains how to set scan options that are used when validating JCLs. Similar to ASG-JOB/SCAN, the ASG-JOB/SCAN Plug-in allows you to specify some options that are used when validating the JCLs. You can set the following options in ASG-JOB/SCAN Plug-in:

## First PROCLIB

When validating JCL, it might be necessary to specify a PROC library to be searched before any other libraries. The Scan Options tab provides a First PROCLIB field for this purpose. When this field is set, JOB/SCAN will look for the PROC and INCLUDE members used in the input JCL file in the library first before searching other libraries. The only type of library supported in this release is PDS.

## Selection Exit

A Selection Exit is a user-defined program that processes the input JCL before JOB/SCAN parses and validates it. The exit evaluates the JCL, makes changes if necessary, and issues a return code instructing JOB/SCAN on the action to be taken with the JCL. The exits can be written in any LE-compliant language. For sample selection exit programs, refer to CA7J1 and CA7J2 members in the SJJSSAMP library. For more information on Selection Exit, see [Selection Exit Interface](#).

## Standards

In addition to the general error validation, the JOB/SCAN's Standards Facility provides the ability to integrate site's additional requirements for validating JCLs. Standards (standards exits or standards routines) are a set of rules that enforces site-specific requirements into JOB/SCAN's JCL validation. The standards are user-defined programs that can be written in any language including OS/VS COBOL, VS COBOL II, REXX, and Assembler. Sample programs are available in the SJJSTDL library. For a detailed introduction to JOB/SCAN's standards facility, see [Standards Information](#).

### *To set the JCL scan options:*

1. Do either of the following to navigate to the **JOB/SCAN** preference page:
  - From the **z/OS Navigator** pane, select (Connect to z/OS) icon. In the **Connect to z/OS** dialog, click **Manage Profiles**.
  - From the **Window** menu, select **Preferences**. In the **Preferences** dialog, select **ASG > JOB/SCAN**.
2. Make sure that the correct **System Name** is selected or add a new one by clicking **Add....**

The screenshot shows the 'Preferences (Filtered)' dialog box for 'System B'. The 'System Name' is 'System B'. The 'Connection Profile' is 'JCL Scan Options'. The fields are filled with: Host URL or IP Address: mvssysb.asg.com; Port Number for z/OSMF: 32008; Port Number for JOB/SCAN: 62000; MVS User ID: myuserid. The 'Apply and Close' button is highlighted with a blue box.

System B profile deleted

System Name: System B

Connection Profile: JCL Scan Options

Host URL or IP Address: mvssysb.asg.com

Port Number for z/OSMF: 32008

Port Number for JOB/SCAN: 62000

MVS User ID: myuserid

Buttons: Add..., Delete, Restore Defaults, Apply, Apply and Close, Cancel

3. Make sure that the **Connection Profile** fields are filled in. See [Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator](#) for more information.
4. Select the **JCL Scan Options** tab and fill the required details.

Preferences (Filtered)

type filter text

ASG

JOB/SCAN

System B profile deleted

System Name: System B

Add... Delete

Connection Profile JCL Scan Options Processing Options

Include summary in JCL scan output

First PROCLIB: ASGJJS.TRY01.PROCLIB

Selection Exit: SEL01

Parameters: PARM24

Standards Name: STD12

Parameters: PARM01

Restore Defaults Apply

Apply and Close Cancel

Field Name	Description
Include summary in JCL scan output	Select to display the JCL scan options in the JCL scan report.
First PROCLIB	Enter the library that are searched first before other libraries for PROC and INCLUDE members.
<b>Selection Exit</b>	
Selection Exit	Enter the name of a selection exit program.
Parameter	Enter any parameters that is required for the execution of the selection exit program.

Field Name	Description
<b>Standards</b>	
Standards Name	Enter the name of a standards program.
Parameter	Enter any parameters that is required for the execution of the standards program.

- 
5. Click **Apply** to save the scan options or click **Apply and Close** to save the scan options and return to the **Connect to z/OS** dialog.

# Setting the Processing Options

This topic explains the procedure to set the processing options that control how JOB/SCAN processes the input JCL, customize various reports and third-party interfaces. These options are defined in the Preferences page. In ASG-JOB/SCAN, user can enter the values for these options on the Processing Options panel or can code in the SJJSPARM library. Values entered on the panel override corresponding parameters coded in the SJJSPARM library. Similarly, values entered in the Plug-in's Preference page will override corresponding parameters coded in JOB/SCAN's SJJSPARM library. You can set the following processing options:

## Use Error Table

JOB/SCAN provides an option to suppress or change the severity level of JCL error messages for a specified JOB/SCAN error numbers. The entries in the Control Table AW are used to suppress any error messages or change their severity level. When the Use Error Table option is selected, JOB/SCAN searches the Control Table AW to determine if any error messages issued on the JCL statement are listed. If the error number is found in Control Table AW, then the severity is changed to the level indicated or is optionally suppressed.

## Condition Code Processing

JOB/SCAN provides an option to control the effect of CONDCODE parameters during execution. The condition code validation is done for COND= parameters on the job statement, EXEC PROC statement, EXEC PGM statements, and for // IF statements. Control Table AU contains Job steps to be ignored by JOB/SCAN. Control Table AR contains the return codes. When the Condition Code Processing option is selected, then JOB/SCAN excludes all program steps identified in Control Table AU entries, and each step is assigned a return code 0 unless a Control Table AR entry is specified.

## Catalog Access

The Catalog Access option controls the access to the system catalog. When this is selected, JOB/SCAN is allowed to access the system catalog.

## VTOC Access

The VTOC Access option controls the access to the DASD VTOCs. When this is selected, JOB/SCAN is allowed to access the DASD VTOCs.

## PDS Member Verification

The PDS Member Verification option determines whether or not JOB/SCAN can validate partitioned dataset member names. Verification is done when a DSN=dsname(membername) parameter is coded in the JCL DD statement.

## Security Exit DSNAME Check

The Security Exit DSNAME Check option allows JOB/SCAN to detect access authority violations. When this option is selected, JOB/SCAN instructs the security exit program to verify that the user ID running JCL validation has the necessary access authority to the datasets that are mentioned in the JCL before actually attempting to access the file.



When this option is selected and if there is no security exit active or no DSNAME-checking support in the exit, then JOB/SCAN ignores this option.

## Program Checking

The Program Checking option determines whether or not JOB/SCAN can validate the existence of programs. If this is selected, JOB/SCAN is allowed to validate the programs.

## Disable SMS Processing

The Disable SMS Processing option determines whether or not JOB/SCAN can access the Automatic Class Selection (ACS) routines and the Construct Access Services (CAS) of the MVS DFSMS subsystem. If this option is selected, JOB/SCAN will bypass all SMS processing. If deselected, the SMS Run option specified in the SJJSPARM library will be in effect.

## IDCAMS Processing

JOB/SCAN has the ability to validate IDCAMS control statements when found within a JCL member. If this option is selected, JOB/SCAN validates IDCAMS control statements and tracks datasets created or deleted by the IDCAMS.

## Use SCHEDULER Interface

When jobs are executed using a scheduler, it allows the JCL to contain environmental variables which must be resolved at the time of job submission. JOB/SCAN supports a wide range of schedulers and it also resolves the environmental variables before the job is submitted. Select this option to turn on the scheduler interface specified for the SCHEDULER Run option in the SJJSPARM library.

## Access Control Cards

A Control Card is a dataset or an inline statement(s) that control how the validation should be processed. When this option is selected, JOB/SCAN is directed to read members within the job stream that contain control cards, verify control card input, and include control cards in the JCL Scan report. You can also limit the number of control cards listed in the JCL Scan view for each DDNAME or dataset name identified as control card input.

## Subsystem

The input JCL may include statements related to Job Entry Subsystem(JES) that need to be processed by JOB/SCAN. This option indicates whether or not JOB/SCAN can validate the JES statements. If JES2 or JES3 is selected, then JOB/SCAN searches for any non-IBM-standard JES statements and uses the entries in Control Table AS to standardize the JES statements.

## Suppress Error Messages

JOB/SCAN provides an option to suppress error messages that are normally displayed in the JCL scan reports. This is done based on the message type. The JOB/SCAN plug-in provides the following options to suppress the messages from being displayed in the JCL Scan view, the Problems view, and the JCL editor:



- **do not suppress**: display all messages
- **advisory only**: suppress advisory messages
- **advisory and warning**: suppress warning and advisory messages
- **advisory, warning, and error**: suppress advisory, warning, and error messages
- **less than or equal to**: suppress all error messages with a severity level equal to or less than the specified value

## Extended Error Text

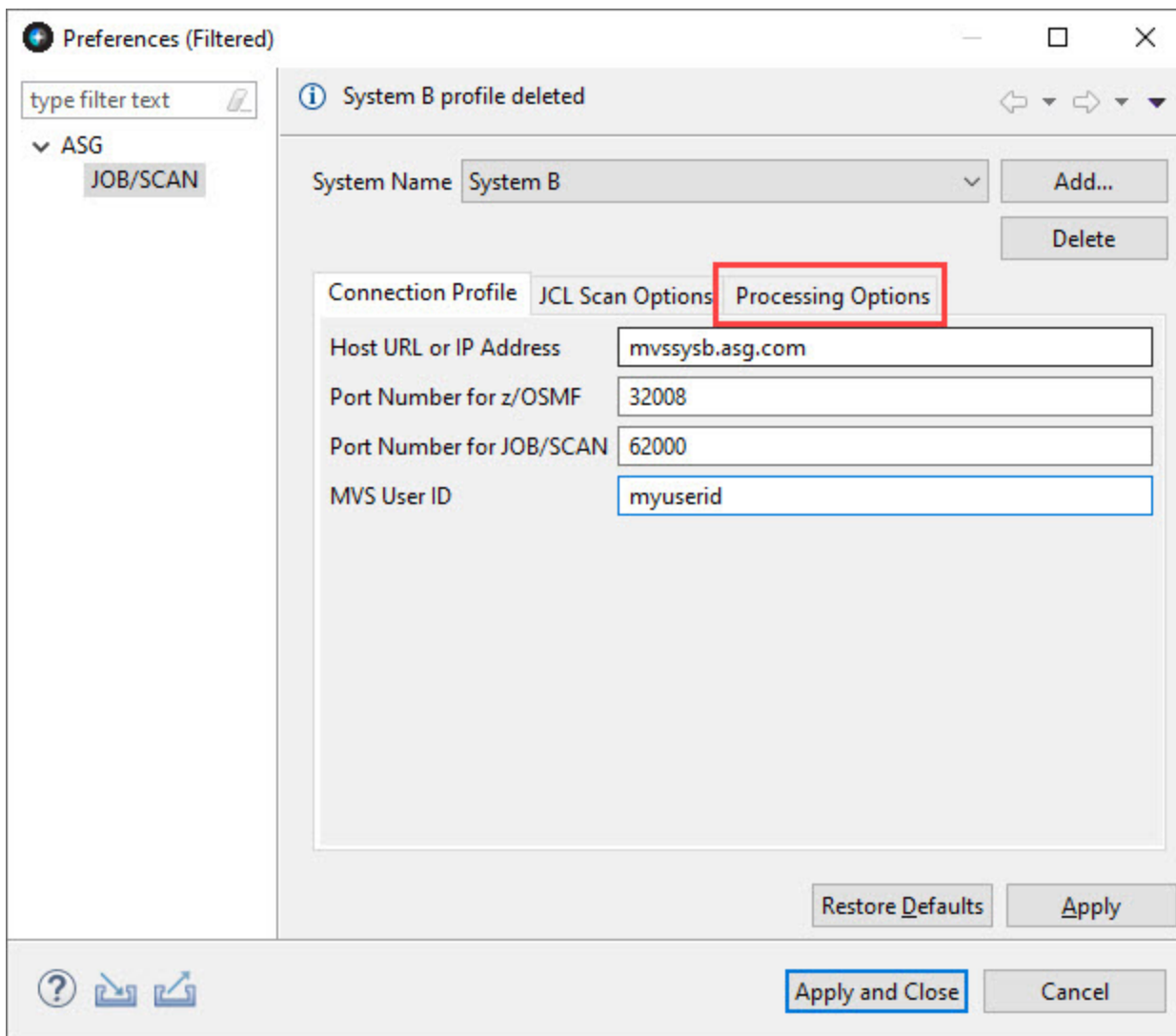
In JOB/SCAN, you can display a messages in two ways: short message text and extended message text. The short error text is a single-line messages that provide less information about the error. The extended messages text give users a better understanding of the violation beyond what the basic short error text provide. This option allows you to select the type of message text to be displayed for each error in the result of a scan.



For more information on processing options, refer to [ASG-JOB/SCAN Reference Information](#).

### *To set the JCL scan options:*

1. Do either of the following to navigate to the **JOB/SCAN** preference page:
  - From the **z/OS Navigator** pane, select (Connect to z/OS) icon. In the **Connect to z/OS** dialog, click **Manage Profiles**.
  - From the **Window** menu, select **Preferences**. In the **Preferences** dialog, select **ASG > JOB/SCAN**.
2. Make sure that the correct **System Name** is selected or add a new one by clicking **Add....**



3. Make sure that the **Connection Profile** fields are filled in. See [Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator](#) for more information.
4. Select the **Processing Options** tab and fill the following details as per your requirement:

Preferences (Filtered)

type filter text

ASG  
JOB/SCAN

System B profile deleted

System Name: System B [Add...]  
[Delete]

Connection Profile | JCL Scan Options | **Processing Options**


Use Error Table  
 Catalog Access  
 PDS Member Verification  
 Program Checking  
 IDCAMS Processing  
 Access Control Cards



Condition Code Processing  
 VTOC Access  
 Security Exit DSNAME Check  
 Disable SMS Processing  
 Use SCHEDULER interface

Control Card Limit: [ ] 1 to 999 or blank for no limit  
Job Limit: [ ] 1 to 999 or blank for no limit  
Subsystem: none [v]  
Suppress Error Messages: do not suppress [v] 0 [ ]  
Extended Error Text: short message text [v]

[Restore Defaults] [Apply]

[?] [ ] [ ] [Apply and Close] [Cancel]

Field Name	Description
Use Error Table	Select to access Control Table AW for error suppression based on severity levels.
Condition Cod Processing	Select to request that JOB/SCAN evaluate all the CONDCODE parameters.
Catalog Access	Deselect to inhibit access to the system catalog.
VTOC Access	Deselect to inhibit access to the DASD VTOCs.
PDS Member Verification	Select to verify the existence of PDS members referenced in the JCL.
	<div style="border: 1px solid purple; padding: 5px; display: inline-block;">  This option is disabled if <b>VTOC Access</b> is not selected.         </div>

Field Name	Description
Security Exit DSNNAME Check	Select to invoke the security exit to check each dataset name in the input JCL for possible security violations.
Program Checking	Select to validate the existence of programs.
Disable SMS Processing	<p>Select to bypass all SMS processing.</p> <div data-bbox="558 495 1487 579" style="border: 1px solid purple; padding: 5px;">  This option has no effect if the SMS subsystem is not active. </div>
IDCAMS Processing	Select to validate IDCAMS control statements.
Use SCHEDULER Interface	<p>Select to turn on the SCHEDULER Run Option in JOB/SCAN's SJJSPARM library.</p> <p>Select to access control card libraries, verify control card input, and include control cards in the validation.</p>
Access Control Cards	<div data-bbox="558 842 1487 961" style="border: 1px solid purple; padding: 5px;">  This option is enabled only if both the <b>Catalog Access</b> and <b>VTOC Access</b> options are selected. </div>
Control Card Limit	Limit the number of control cards (1-999) listed in the JCL Scan view for each DDNAME or dataset name identified as control card input.
Job Limit	<div data-bbox="558 1094 1487 1213" style="border: 1px solid purple; padding: 5px;">  This field is enabled only if <b>Access Control Cards</b> is enabled and selected. </div>
Subsystem	Limit the number of jobs to be processed (1-999) in each JCL scan.
Suppress Error Messages	<p>Use <b>JES2</b>, <b>JES3</b>, or <b>none</b> to indicate whether the JCL contains JES2 or JES3 statements to be validated for syntax. JES statements are not evaluated if <b>none</b> is selected for this option.</p> <p>Select the type of messages to suppress in the result of a scan, including the JCL Scan view, the Problems view, and the JCL editor. The options are:</p> <ul style="list-style-type: none"> <li>• do not suppress</li> <li>• advisory only</li> <li>• advisory and warning</li> <li>• advisory, warning, and error</li> <li>• less than or equal to</li> </ul>

Field Name	Description
Severity Level	Select the severity level below or at which error messages are suppressed. This selection is enabled only if <b>less than or equal to</b> is selected for <b>Suppress Error Messages</b> option.
Extended Error Text	Select the type of message text to be displayed for each error in the result of a scan. The options are: <ul style="list-style-type: none"><li data-bbox="673 506 927 537">• short message text</li><li data-bbox="673 554 976 585">• extended message text</li></ul>

- 
5. Click **Apply** to save the scan options or click **Apply and Close** to save the scan options and return to the **Connect to z/OS** dialog.

## Working with z/OS Datasets


This topic explains how to:

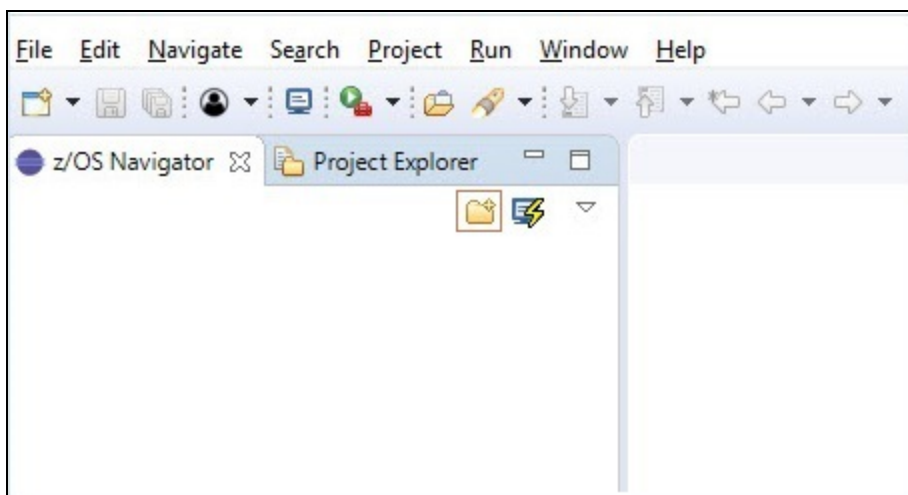
- displays the z/OS datasets
- edit the z/OS datasets
- refresh the list of datasets

### Displaying Datasets

Datasets that are in the z/OS system can be explored either from z/OS Navigator or Remote Systems view.

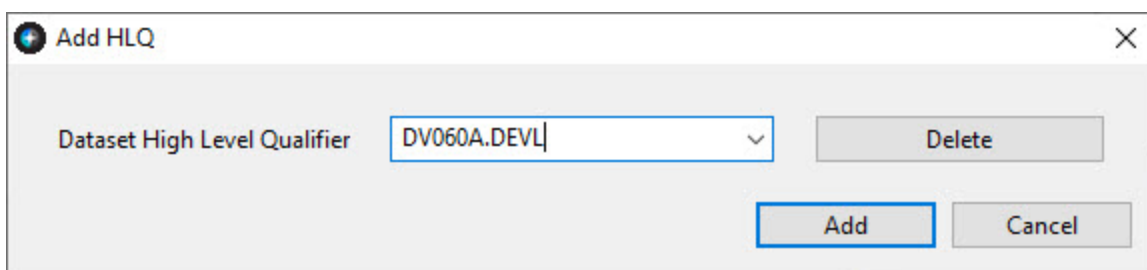
**To display datasets in z/OS Navigator:**

1. Select  (Add High Level Qualifier) icon from the z/OS Navigator view. The ADD HLQ dialog box appears.



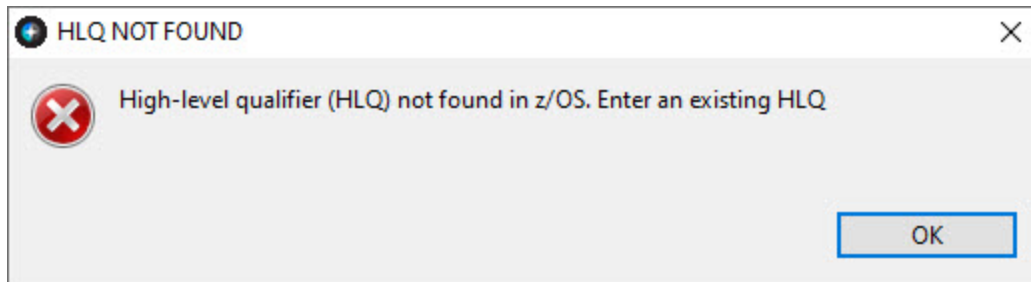
If connection to z/OS is not established, then **Connect to z/OS** dialog appears and prompts for main-frame credentials. See [Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator](#) for more information.

2. In the **Dataset High Level Qualifier**, enter the dataset's high-level qualifier that you would like to view and click **Add**. Datasets are displayed under a High-Level Qualifier (HLQ) in a tree-like structure.

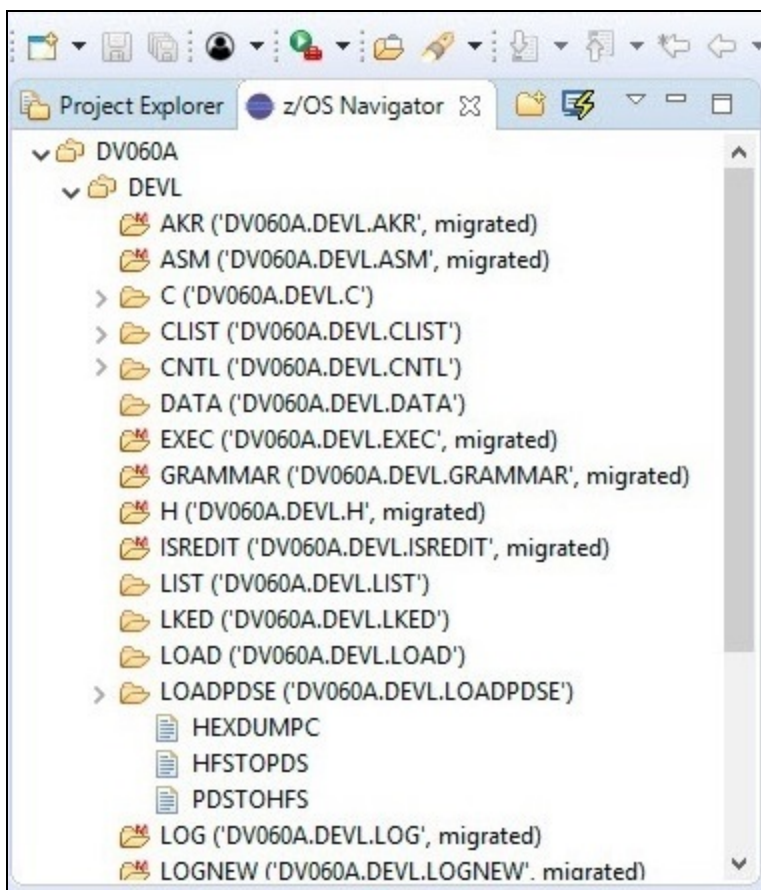










- If you have entered the high-level qualifier for the first time, then on clicking **Add**, the HLQ is saved to the **Dataset High Level Qualifier** list. The list keeps track of the HLQs that you have entered in the past so you don't have to retype again.
- To delete a saved HLQ from the **Dataset High Level Qualifier** list, select the HLQ and click **Delete**.
- If the entered HLQ is not found in z/OS, then the following error message is displayed. Make sure to enter an existing HLQ.



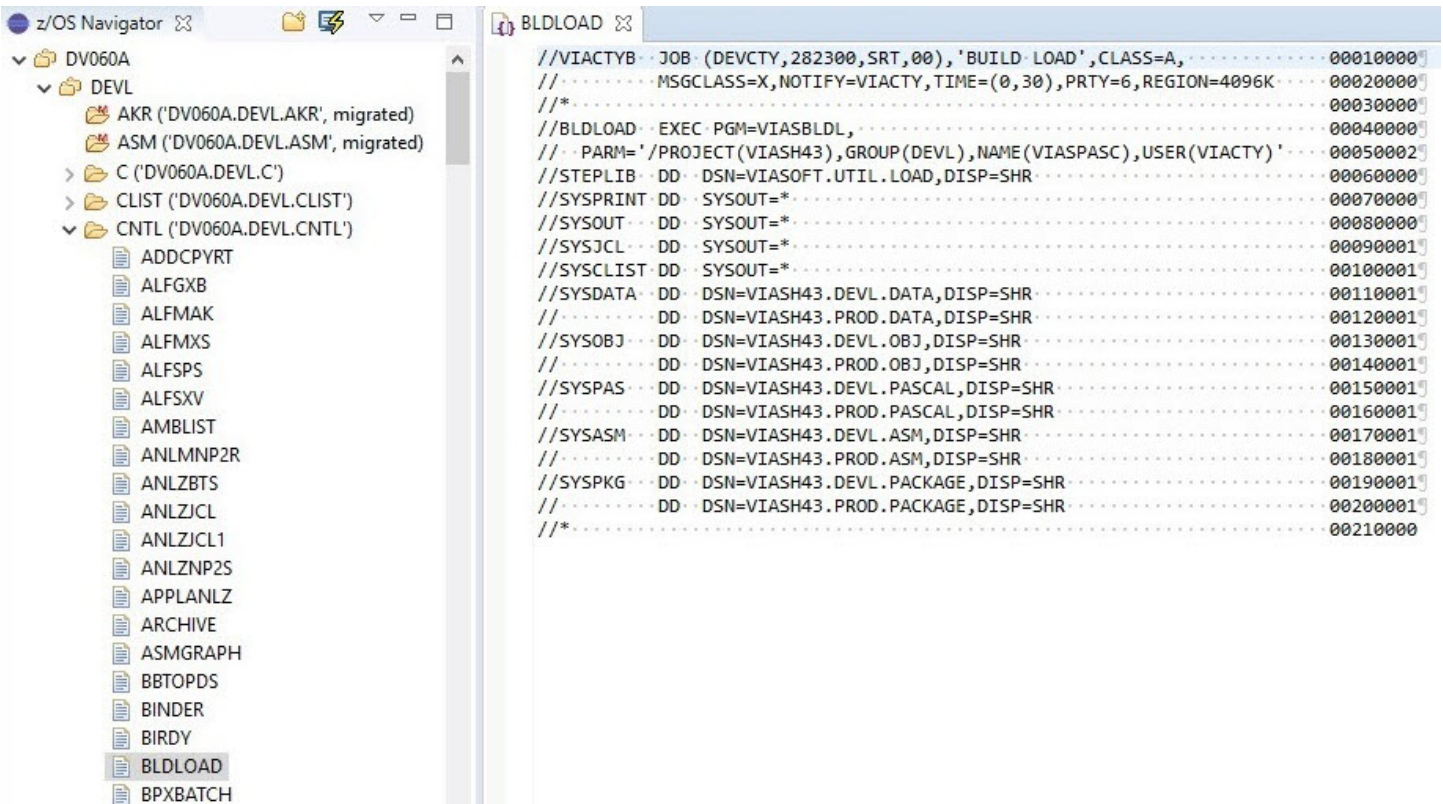
3. Expand the tree-like structure to view the files.



Icons in the tree structure represent the following:

Icon	Description
	node in a dataset qualifier
	PDS or PDSE containing dataset members
	sequential dataset
	partitioned dataset that has been migrated
	sequential dataset that has been migrated
	member in PDS or PDSE

4. Double-click on a dataset, the z/OS navigator opens the file in the editor. A tab in the editor displays the name of the file, while its contents appear in the main region of the tabbed pane.



The screenshot shows the z/OS Navigator interface. On the left, a tree view displays the dataset structure for DV060A, including a DEVL sub-qualifier with members like AKR, ASM, C, CLIST, and CNTL. The CNTL member is expanded to show a list of dataset members, with BLDLOAD selected. On the right, the BLDLOAD dataset is open in the editor, displaying its JCL content. The JCL includes job parameters, dataset definitions (DD), and system output (SYSOUT) specifications.

```

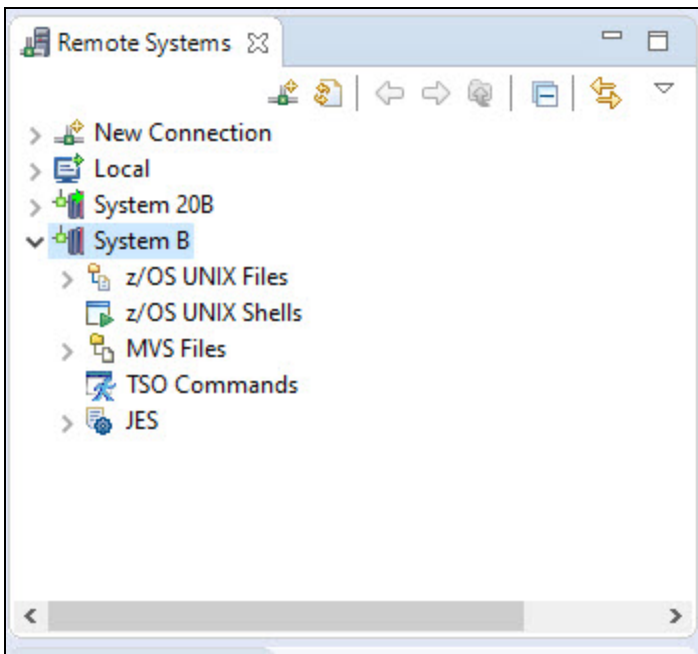
//VIACTYB JOB (DEVCTY,282300,SRT,00), 'BUILD LOAD',CLASS=A, ..... 00010000
//..... MSGCLASS=X,NOTIFY=VIACTY,TIME=(0,30),PRTY=6,REGION=4096K ..... 00020000
//* ..... 00030000
//BLDLOAD EXEC PGM=VIASBLDL, ..... 00040000
// PARM='/PROJECT(VIASH43),GROUP(DEVL),NAME(VIASPASC),USER(VIACTY)' ..... 00050002
//STEPLIB DD DSN=VIASOFT.UTIL.LOAD,DISP=SHR ..... 00060000
//SYSPRINT DD SYSOUT=* ..... 00070000
//SYSOUT DD SYSOUT=* ..... 00080000
//SYSJCL DD SYSOUT=* ..... 00090001
//SYSCLIST DD SYSOUT=* ..... 00100001
//SYSDATA DD DSN=VIASH43.DEVL.DATA,DISP=SHR ..... 00110001
//..... DD DSN=VIASH43.PROD.DATA,DISP=SHR ..... 00120001
//SYSOBJ DD DSN=VIASH43.DEVL.OBJ,DISP=SHR ..... 00130001
//..... DD DSN=VIASH43.PROD.OBJ,DISP=SHR ..... 00140001
//SYSPAS DD DSN=VIASH43.DEVL.PASCAL,DISP=SHR ..... 00150001
//..... DD DSN=VIASH43.PROD.PASCAL,DISP=SHR ..... 00160001
//SYSASM DD DSN=VIASH43.DEVL.ASM,DISP=SHR ..... 00170001
//..... DD DSN=VIASH43.PROD.ASM,DISP=SHR ..... 00180001
//SYSPKG DD DSN=VIASH43.DEVL.PACKAGE,DISP=SHR ..... 00190001
//..... DD DSN=VIASH43.PROD.PACKAGE,DISP=SHR ..... 00200001
//* ..... 00210000

```

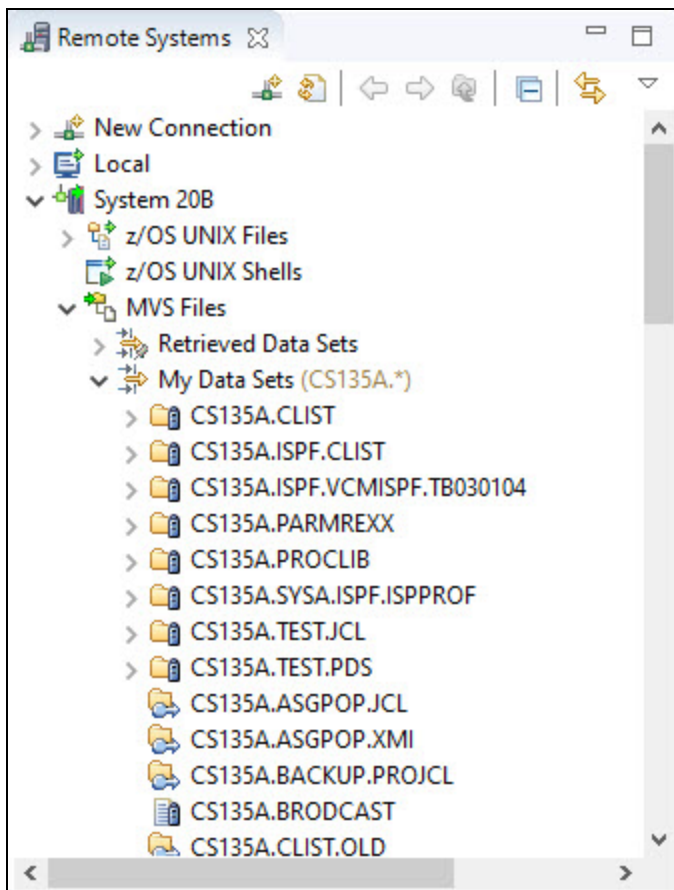
#### To display datasets in Remote System:

1. In the **Remote Systems** view, expand your z/OS connection.



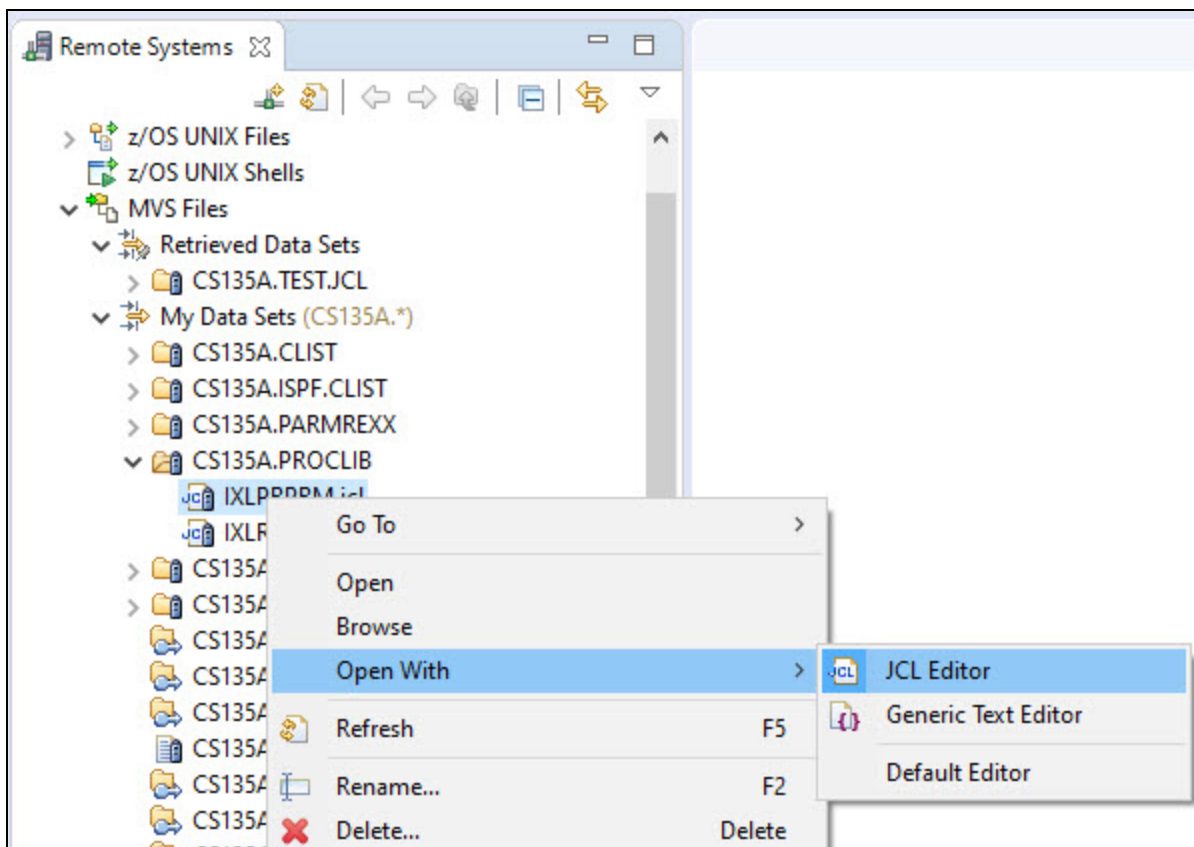


2. Navigate to **MVS Files > My Data Sets**. z/OS Explorer displays all the datasets with your TSO ID as their high-level qualifier.



3. Right-click on a dataset and select **Open With --> JCL Editor**.

★ If you don't find the **JCL Editor** option, select **Other...**, and in the **Editor Selection** dialog, select **JCL Editor** from the **Internal Editors** list.



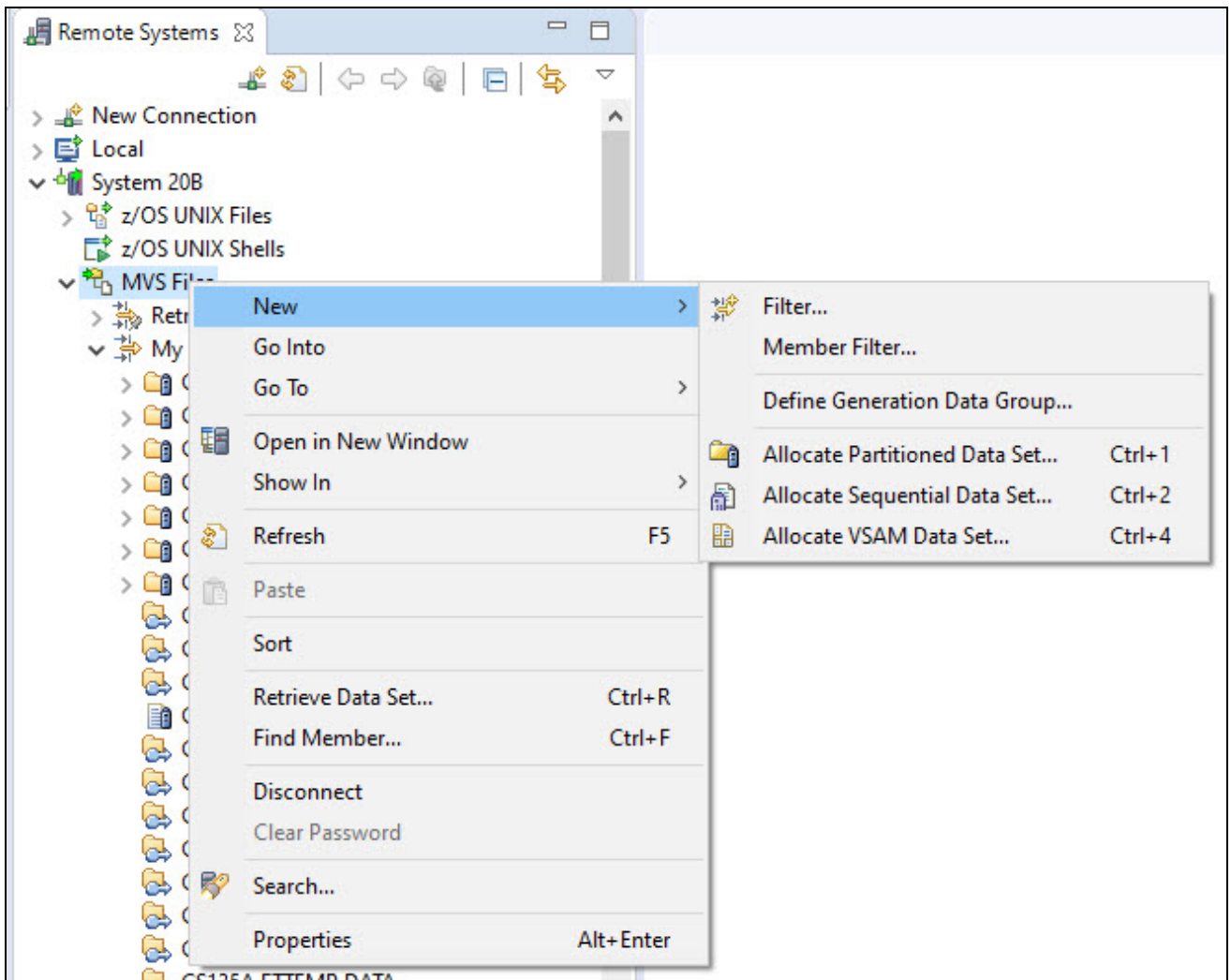
★ When a file is opened from Remote Systems view, JOB/SCAN commands are available in JCL Editor **only**. The commands are displayed as buttons in the toolbar as well as additional options in the Edit menu. If you are not already connected to a JOB/SCAN server when you run the JOB/SCAN commands, you will be asked to connect to JOB/SCAN first before the commands are executed. Perform the steps in [Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator](#) to connect to JOB/SCAN's server.

6. The dataset is displayed in the JCL Editor.

★ If you want to access datasets with high-level qualifiers other than your TSO ID, then right-click on **MVS Files** and select on of the following options:



- New > Filter or Member Filter
- Retrieve Datasets
- Search



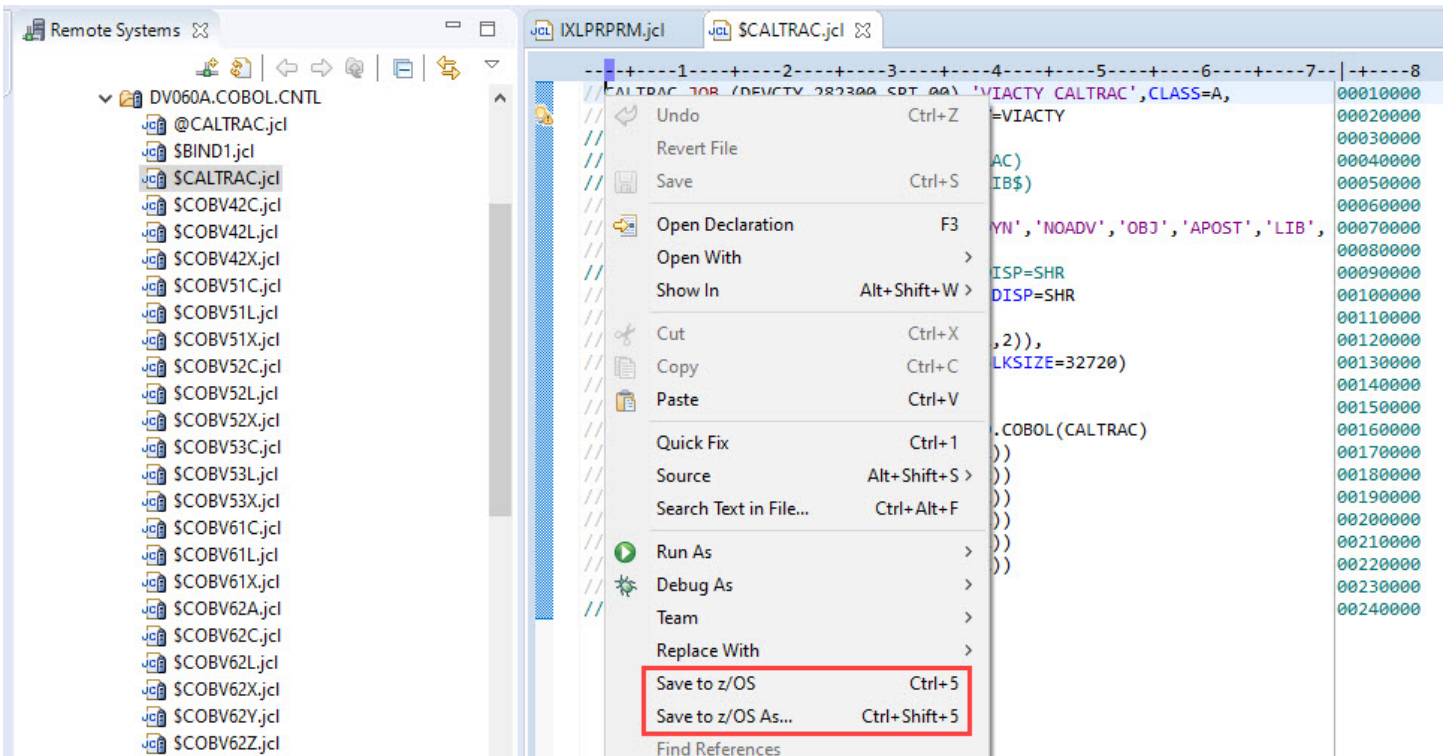
For more information about these options, see [IBM Explorer for z/OS V3.2 documentation](#).

## Editing and Saving Datasets

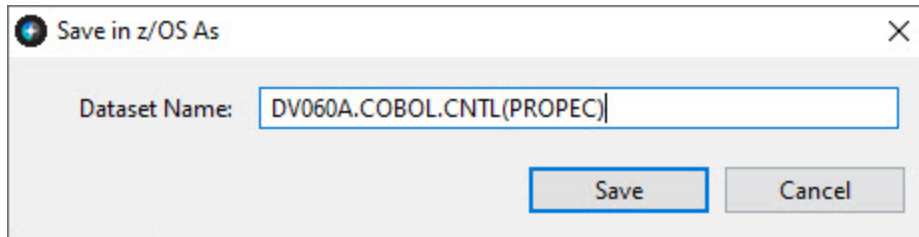
The JCL Editor allows the user to edit and save a dataset to z/OS.

### **To edit and save a dataset using z/OS Navigator:**

1. In the **z/OS Navigator**, double-click on a dataset that you like to update. The dataset is displayed in the editor area.
2. After updating the dataset, right-click in the JCL editor area to bring up the dataset contextual menu.



3. Select one of the options to save the changes to z/OS server:
  - a. **Save to z/OS** - This option allows the user to save the file under the same dataset name.
  - b. **Save to z/OS As...** - This option allows the user to save the file either as a new member in a PDS or as an existing PS file. When this option is selected, user is prompted to enter a dataset name. Enter a name and click **OK**.



★ This option will not allocate a new dataset.

4. Refresh the z/OS Navigator to view the changes.

#### **To edit and save a dataset using Remote Systems:**

1. In the **Remote Systems**, right-click on a dataset that you like to update and select **Edit With --> JCL Editor**. The dataset is displayed in the **JCL Editor**.

2. After updating the dataset, do one of the following:

- Press **Ctrl+ S**
- Select **File > Save** or **Save As**
- Right-click on the file and select **Save**.

When you save the changes, the dataset on the z/OS server is automatically updated.

3. You may have to refresh to view the changes.

## Refreshing Dataset

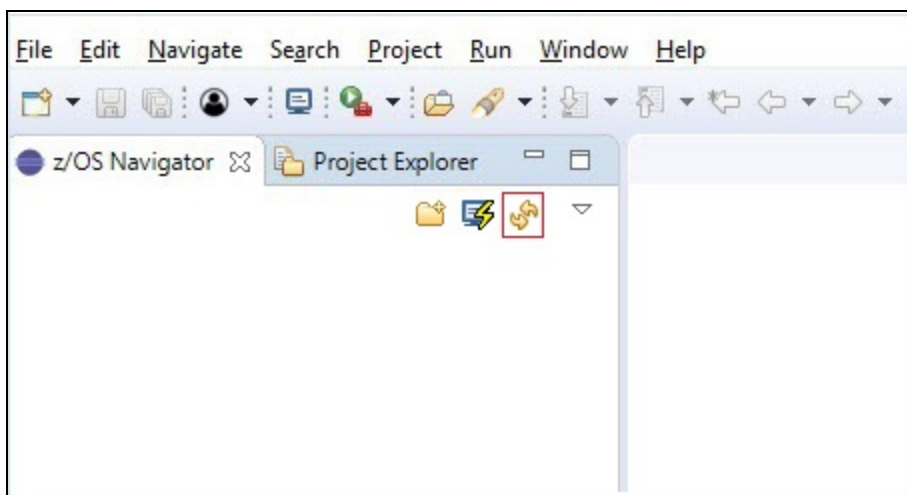
The z/OS Navigator and Remote Systems provides a refresh icon that refreshes the list of datasets.

You may need to refresh dataset list whenever you:

- save a file to z/OS
- add / delete / edit a dataset or a member in z/OS

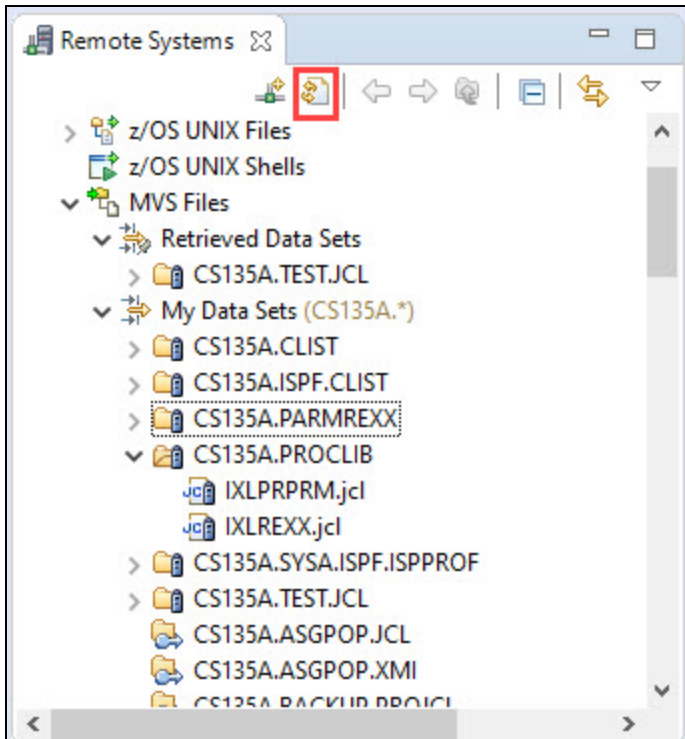
**To refresh datasets in z/OS Navigator:**

- Click the 🔄 (Refresh View) icon in the **z/OS Navigator** view.



**To refresh datasets in Remote Systems:**

- In the **Remote Systems** view, select **MVS Files** or the files / folder under the MVS Files and click the 🔄 (Refresh information of selected resource) icon.



See [Validating JCL](#) for information on scanning / validating a file.

# Working with Local Files

ASG-JOB/SCAN Plug-in allows the user to scan files that are present in your local file system. This topic explains how to work with the files that are in your local file system.

## Displaying Local Files

Files in your local file system can be explored either from Remote Systems view or Project Explorer view.

### Displaying files in Project Explorer

To display local files in the Project Explorer view, first they must be imported. You can import the files by one of the following methods:

- dragging and dropping from the file system
- copying and pasting from the file system
- using the Import option

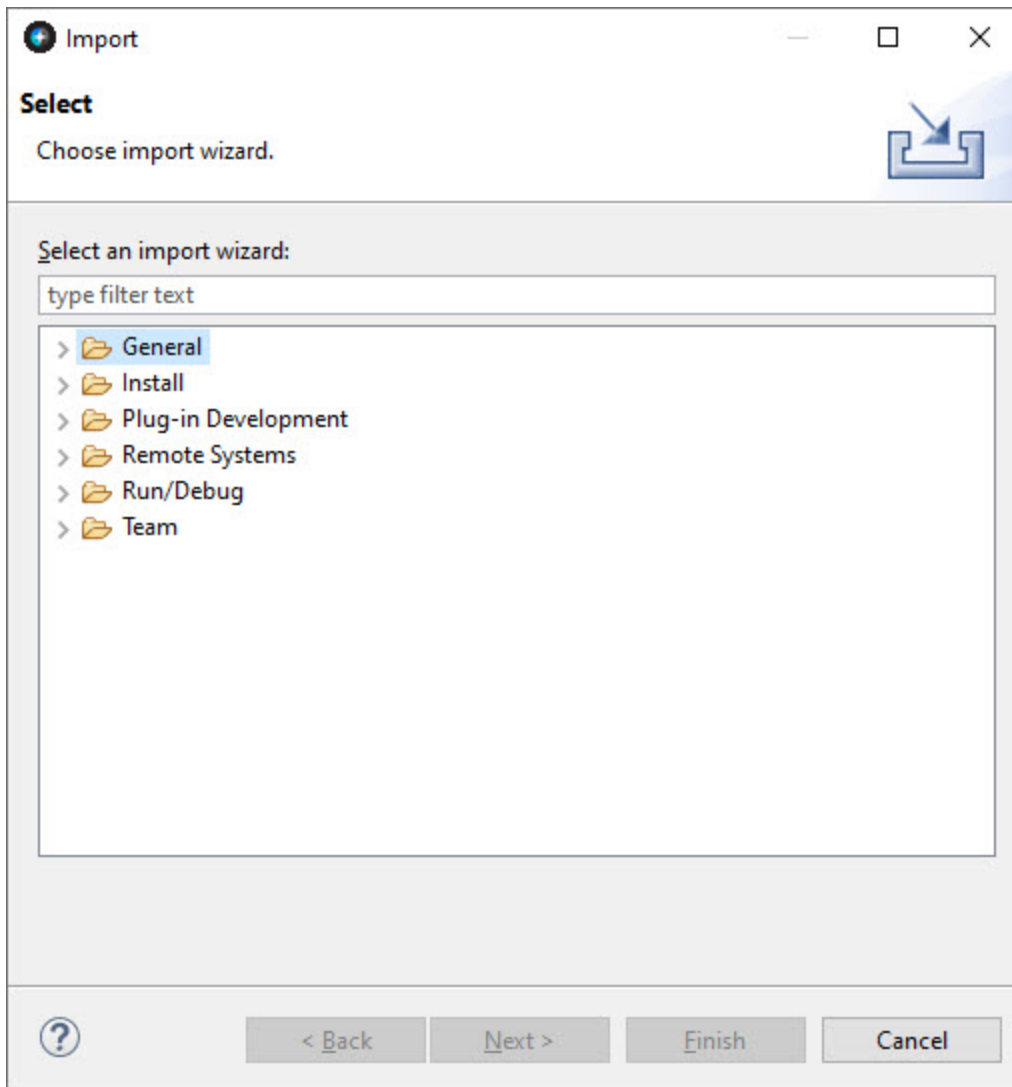
After importing the files, right-click on the file that you like to view and select **Open With --> JCL Editor**.



- If you don't find the **JCL Editor** option, select **Other...** and in the **Editor Selection** dialog, select **JCL Editor** from the **Internal Editors** list.
- When a file is opened from **Project Explorer** view, JOB/SCAN commands are available in JCL Editor **only**. The commands are displayed as buttons in the toolbar as well as additional options in the **Edit** menu. If you are not already connected to a JOB/SCAN server when you run the JOB/SCAN commands, you will be asked to connect to JOB/SCAN first before the commands are executed. Perform the steps in [Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator](#) to connect to JOB/SCAN's server.

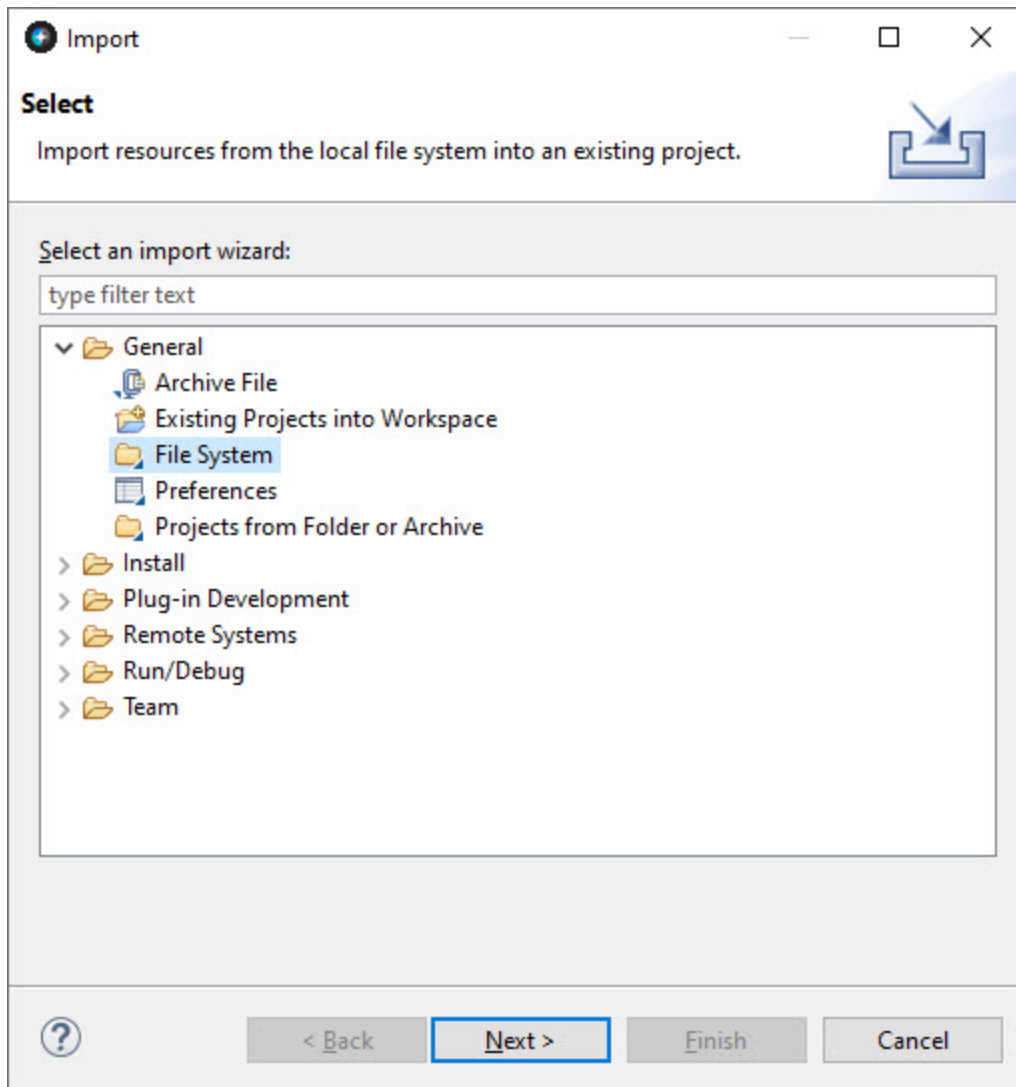
### *To import files using Import option:*

1. From the main menu, select **File > Import...** The **Import** dialog appears.

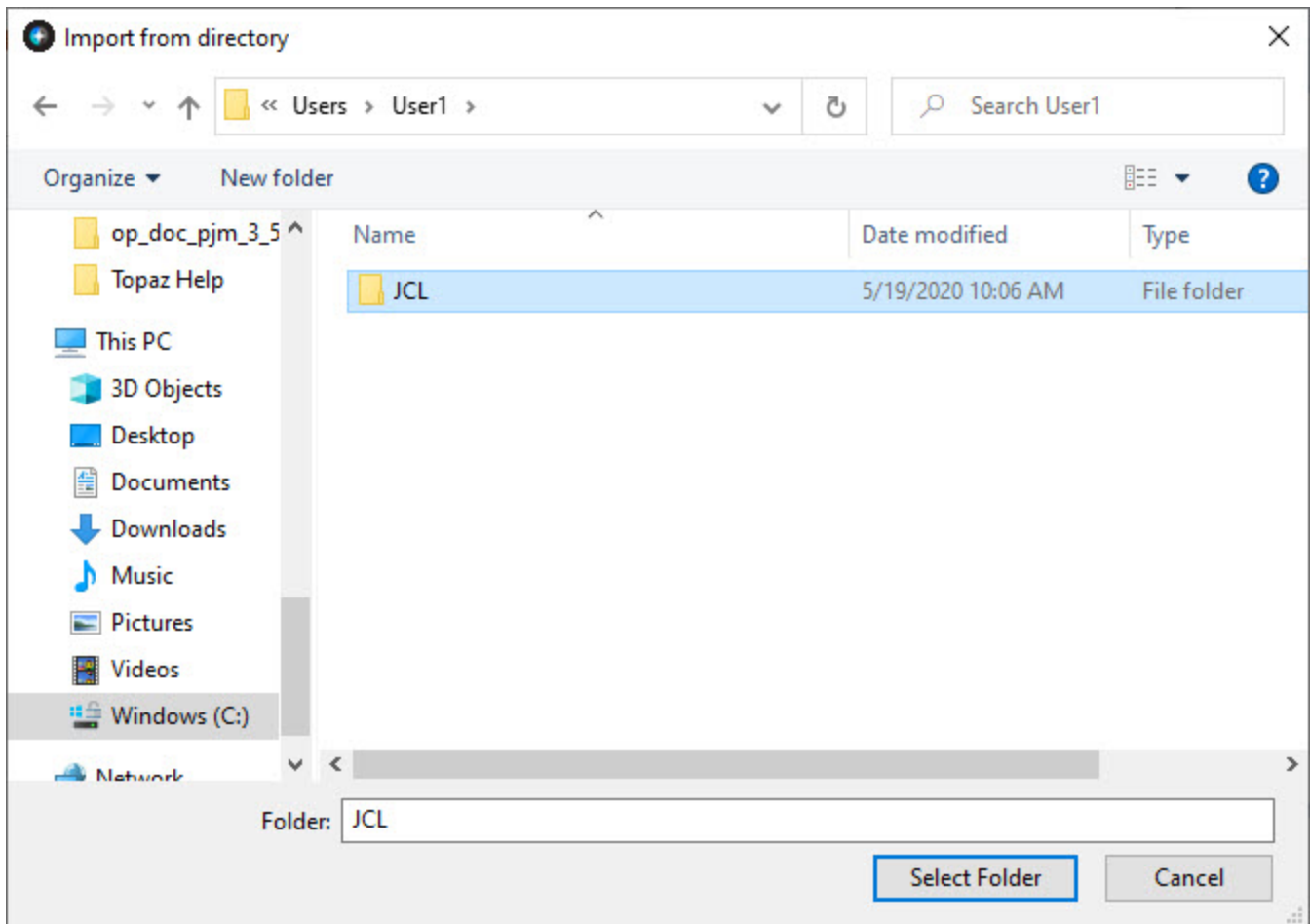


2. Expand the **General** folder, select **File System** and click **Next**.

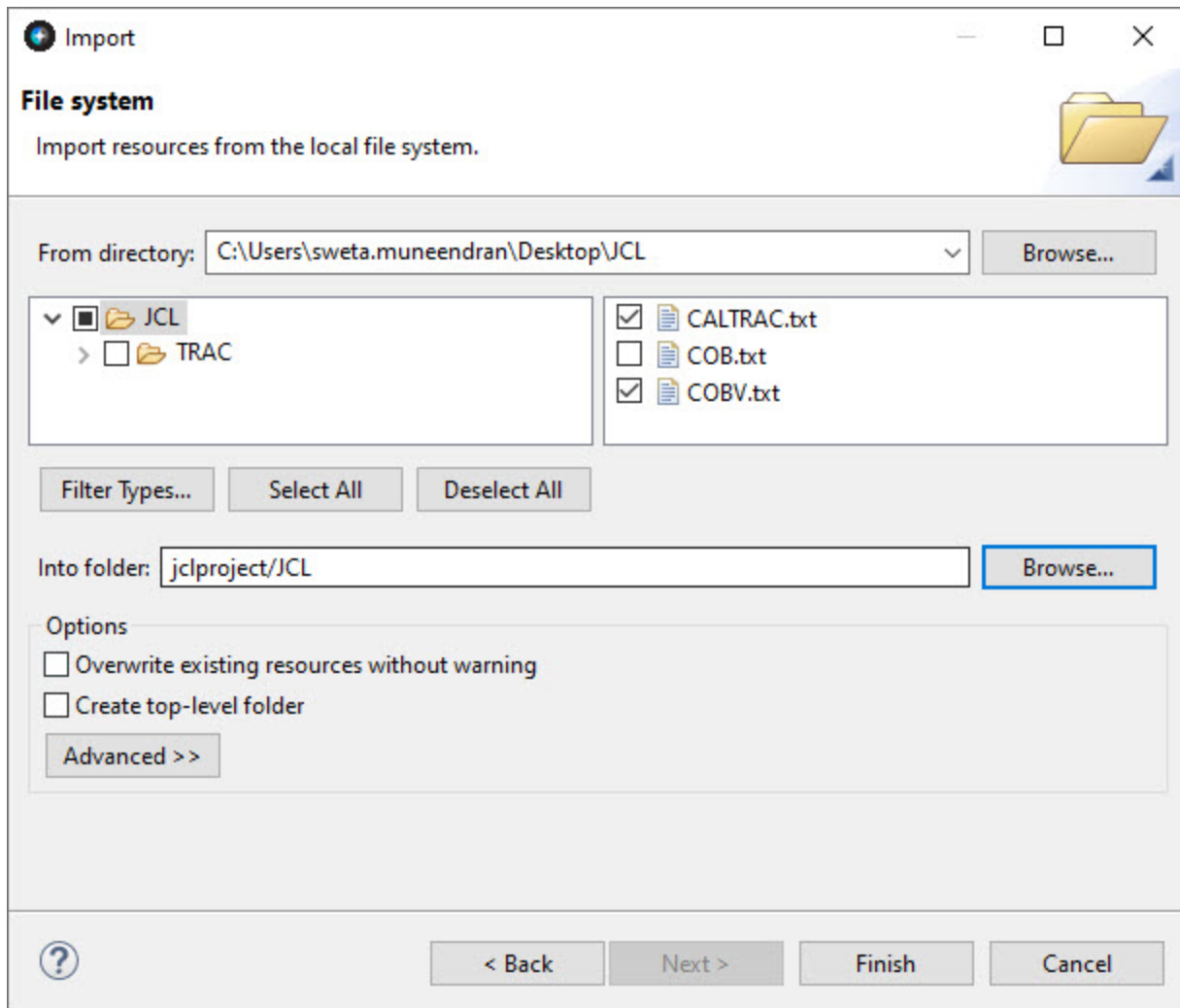




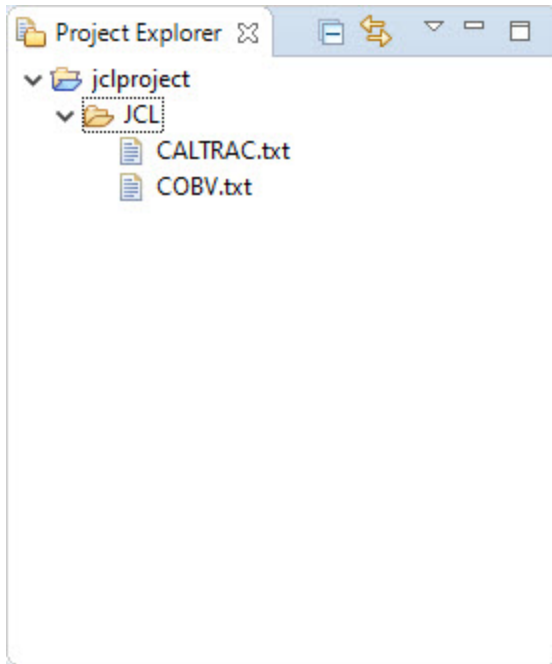
3. In the next page of the Import dialog, click the **Browse** button. The **Import from directory** dialog appears.



4. Select the folder from which you would like to add the files and click **Select Folder**. The directory is displayed in the left pane of the **Import** dialog.



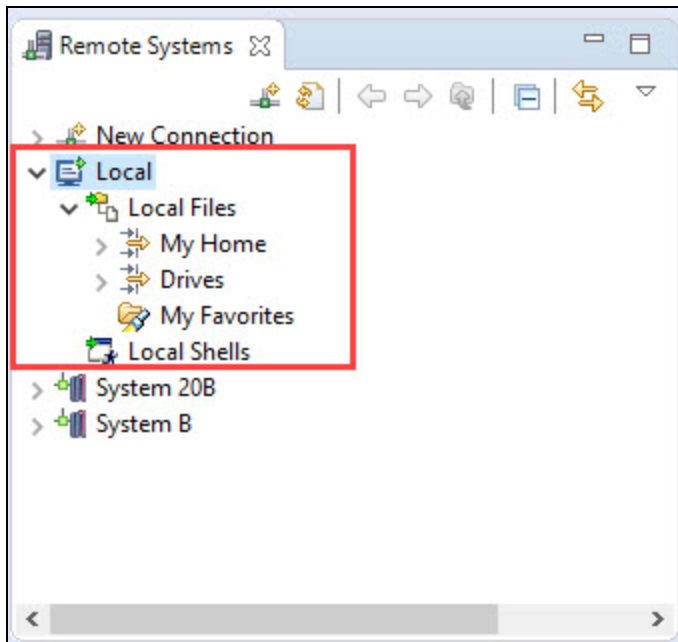
5. To select the files you want to add, expand the hierarchies in the left pane and select or clear the checkboxes that represent the folders in the selected directory. Then in the right pane, select or clear checkboxes for individual files.
6. In the **Into folder** option, specify the project or folder that will be the import destination.
7. Click **Finish**. The files are displayed in the **Project Explorer** view under the specified destination.



### Displaying files in Remote Systems

#### To display files in Remote Systems:

1. In the **Remote Systems** view, expand **Local > Local Files**. The locations of your local file system is displayed under two nodes: **My Home** and **My Drives**.



2. Select the file system that you want and continue navigating until you find the file you are looking for.
3. Right-click on the file and select **Open With --> JCL Editor**.



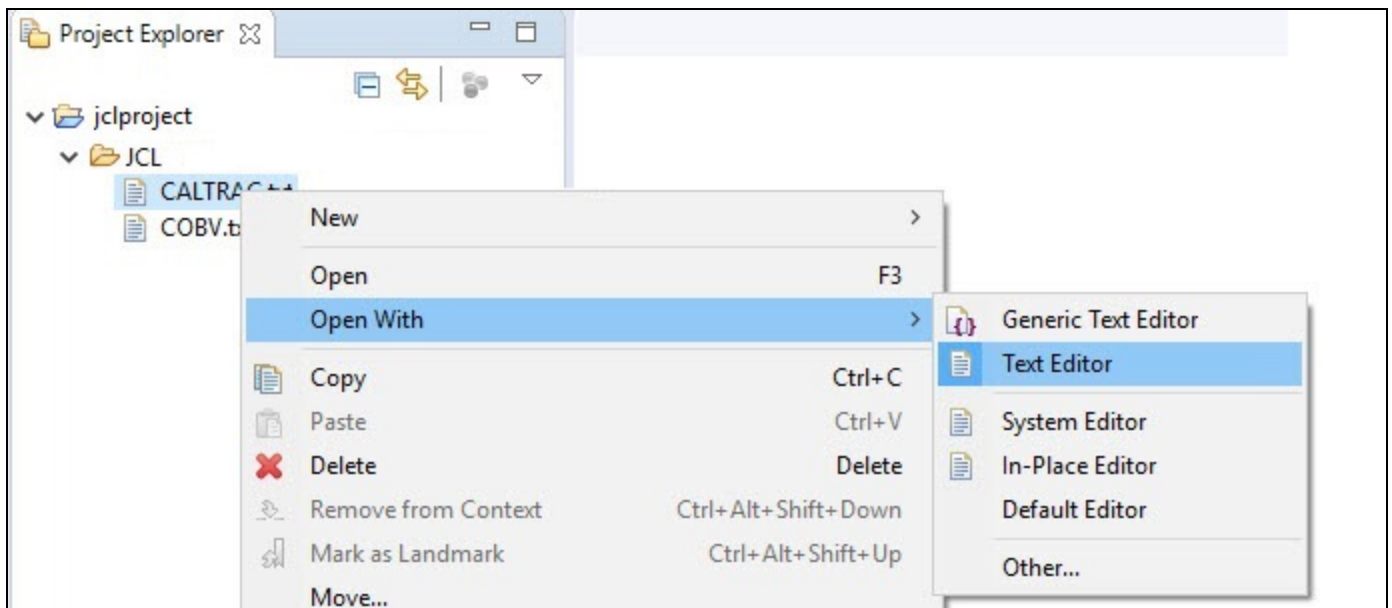
- If you don't find the **JCL Editor** option, select **Other...** and in the **Editor Selection** dialog, select **JCL Editor** from the **Internal Editors** list.
- When a file is opened from **Remote Systems** view, JOB/SCAN commands are available in JCL Editor **only**. The commands are displayed as buttons in the toolbar as well as additional options in the **Edit** menu. If you are not already connected to a JOB/SCAN server when you run the JOB/SCAN commands, you will be asked to connect to JOB/SCAN first before the commands are executed. Perform the steps in [Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator](#) to connect to JOB/SCAN's server.

## Editing and Saving Local Files

The JCL Editor allows you to edit, save and scan a file that is in your local file system.

### To edit a file:

1. Select a file in the **Project Explorer** view or **Remote Systems** view.
2. Right-click on the file and select **Open With > JCL Editor**. The file is displayed in the editor area.



If you don't find the **JCL Editor** option, select **Other...**, and in the **Editor Selection** dialog, select **JCL Editor** from the **Internal Editors** list.

3. After updating the file, do one of the following:
  - Press **Ctrl+ S**
  - Select **File > Save** or **Save As**
  - Right-click on the file and select **Save**.



- If the file opens in an external editor outside the z/OS Explorer workbench, then you probably opened the file without selecting an editor.
- ASG recommends you to open the file in JCL Editor.
- You may not see the JOB/SCAN functions if you have opened the file in different editor other than JCL Editor.

See [Validating JCL](#) for information on scanning / validating a file.

# Validating JCL

ASG-JOB/SCAN validates JCL in an input job stream, so that potential errors, including run-time, hidden, site-specific and syntax errors can be corrected before the job stream is submitted for processing. Similarly, ASG-JOB/SCAN Plug-in processes the entire job stream in the z/OS environment and displays the results in the JCL Scan and Problems views.

You can validate a JCL file using the following options:

- [Scan JCL](#)
- [Scan and submit JCL](#)


## Scan JCL

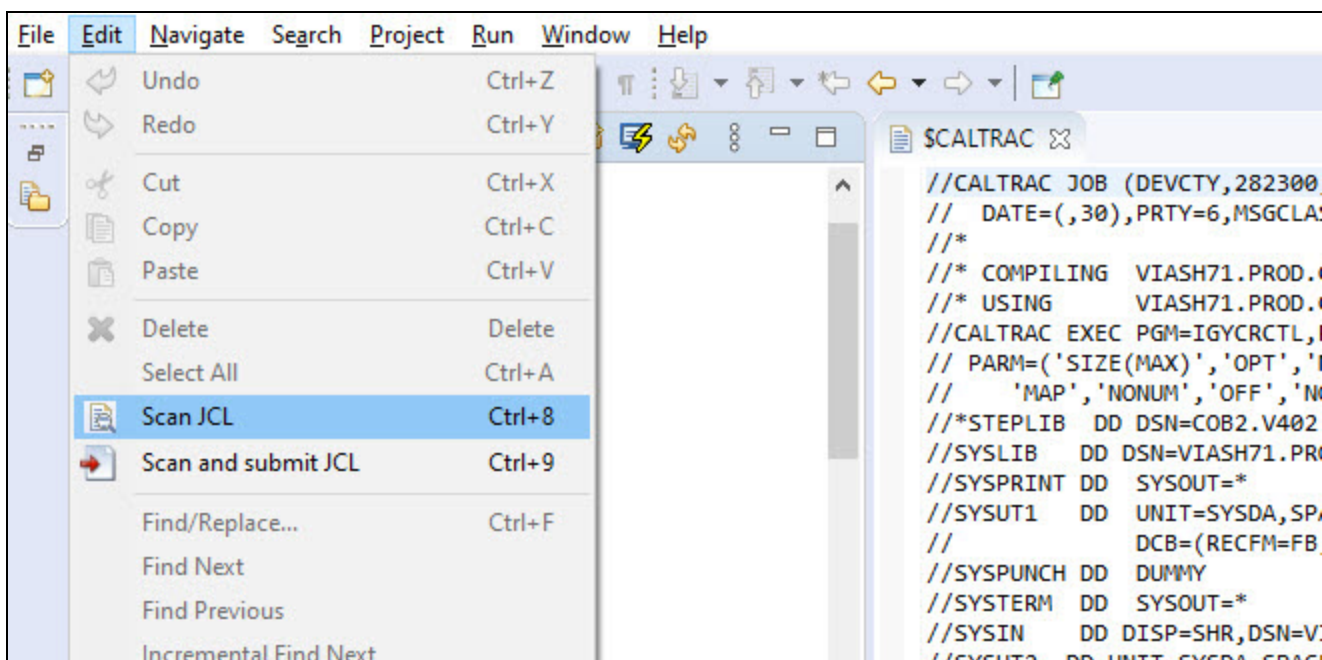
Scan JCL option available in the ASG-JOB/SCAN Plug-in is similar to ASG-JOB/SCAN's JEM and JSCAN edit macros. The edit macros validates the JCL file during an ISPF Edit session and displays the Structured JCL Listing (SJL) report. Similarly, the Scan JCL option validates the JCL file that is active in the JCL Editor and displays the results in JCL Scan and Problems views.



- If you are not connected to a JOB/SCAN server when you run the JOB/SCAN commands, you will be asked to connect to JOB/SCAN first before the commands are executed. Perform the steps in [Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator](#) to connect to JOB/SCAN's server.
- When a file is opened from **Remote Systems** or **Project Explorer** view, JOB/SCAN commands are available in JCL Editor **only**.
- JCL scan options and processing options must be specified before you begin to scan a JCL file. For more information on setting these options, see [Setting the JCL Scan Options](#) and [Setting the Processing Options](#).

### To scan JCL:

1. Open the file that you want to scan in JCL Editor.
2. Make changes, if any, and save it.
3. Select **Scan JCL** option from the **Edit** menu or  icon from the toolbar.



4. JOB/SCAN Plug-in validates the input JCL and displays the reports in the **JCL Scan** and **Problems** views. For more information, refer to [Reports](#).



When you scan another file in the JCL editor, the report in the JCL Scan view is replaced. However, the messages in the Problems view are not replaced, instead they add up to the existing messages.



The screenshot shows the z/OS Navigator interface. On the left, a file tree displays the directory structure under 'CNTL (DV060A.COBOL.CNTL)', with 'SCALTRAC' selected. The main window shows the JCL code for 'SCALTRAC'. Two lines are highlighted with red dashed lines and red 'X' markers in the left margin, indicating errors: the 'DATE' parameter and the 'SYSLIB' DSN. Below the JCL editor, the 'Problems' pane shows '2 errors, 0 warnings, 1 other'. The 'JCL Scan' pane displays the error messages: 'DSS1080E - INVALID JOB PARAMETER - "DATE"' and 'DSS4050E - DATA SET NOT FOUND IN CATALOG'.

```

//CALTRAC JOB (DEVCTY,282300,SRT,00), 'VIACTY CALTRAC', CLASS=A,      00010000
// DATE=(,30), PRTY=6, MSGCLASS=X, NOTIFY=VIACTY                    00020000
//*                                                                    00030000
//* COMPILING VIASH71.PROD.COBOL(CALTRAC)                            00040000
//* USING      VIASH71.PROD.COBOL($MACLIB$)                          00050000
//CALTRAC EXEC PGM=IGYCRCTL, REGION=2M,                               00060000
// PARM=('SIZE(MAX)', 'OPT', 'NORES', 'NODYN', 'NOADV', 'OBJ', 'APOST', 'LIB', 00070000
//   'MAP', 'NONUM', 'OFF', 'NOOBJ')                                  00080000
//*STEPLIB DD DSN=COB2.V402.COB2COMP, DISP=SHR                       00090000
//*SYSLIB DD DSN=VIASH71.PROD.COPYLIB, DISP=SHR                      00100000
//SYSPRINT DD SYSOUT=*                                               00110000
//SYSUT1 DD UNIT=SYSDA, SPACE=(CYL,(1,2)),                          00120000
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=32720)                          00130000
//SYSPUNCH DD DUMMY                                                 00140000
//SYSTEM DD SYSOUT=*                                                00150000
//SYSIN DD DISP=SHR, DSN=VIASH71.PROD.COBOL(CALTRAC)                00160000
//SYSOUT DD UNIT=SYSDA, SPACE=(CYL,(1,2))                            00170000
//SYSUT2 DD UNIT=SYSDA, SPACE=(CYL,(1,2))                            00180000
//SYSUT3 DD UNIT=SYSDA, SPACE=(CYL,(1,2))                            00190000
//SYSUT4 DD UNIT=SYSDA, SPACE=(CYL,(1,2))                            00200000
//SYSUT5 DD UNIT=SYSDA, SPACE=(CYL,(1,2))                            00210000
//SYSUT6 DD UNIT=SYSDA, SPACE=(CYL,(1,2))                            00220000
//SYSUT7 DD UNIT=SYSDA, SPACE=(CYL,(1,2))                            00230000
//SYSI TN DD DUMMY                                                  00230000

```

Problems: 2 errors, 0 warnings, 1 other

Description	Resource
> Errors (2 items)	
> Infos (1 item)	

```

//CALTRAC JOB (DEVCTY,282300,SRT,00), 'VIACTY CALTRAC', CLASS=A,      00010000
// DATE=(,30), PRTY=6, MSGCLASS=X, NOTIFY=VIACTY                    00020000
***ERROR - DSS1080E - INVALID JOB PARAMETER - "DATE" .              00030000
//*                                                                    00040000
//* COMPILING VIASH71.PROD.COBOL(CALTRAC)                            00050000
//* USING      VIASH71.PROD.COBOL($MACLIB$)                          00060000
//CALTRAC EXEC PGM=IGYCRCTL, REGION=2M,                               00070000
// PARM=('SIZE(MAX)', 'OPT', 'NORES', 'NODYN', 'NOADV', 'OBJ', 'APOST', 'LIB', 00080000
//   'MAP', 'NONUM', 'OFF', 'NOOBJ')                                  00090000
//*STEPLIB DD DSN=COB2.V402.COB2COMP, DISP=SHR                       00100000
//*SYSLIB DD DSN=VIASH71.PROD.COPYLIB, DISP=SHR                      00110000
***ERROR - DSS4050E - DATA SET NOT FOUND IN CATALOG                00120000

```

- After validation, JOB/SCAN Plug-in displays markers on the right and left margins of the JCL Editor. These markers help you to quickly identify a problem in a file. See [Markers](#) for more information.

This close-up screenshot of the JCL editor shows the same code as above. Red dashed lines and red 'X' markers in the left margin highlight the 'DATE' and 'SYSLIB' lines, indicating the specific errors.

```

//CALTRAC JOB (DEVCTY,282300,SRT,00), 'VIACTY CALTRAC', CLASS=A,      00010000
// DATE=(,30), PRTY=6, MSGCLASS=X, NOTIFY=VIACTY                    00020000
//*                                                                    00030000
//* COMPILING VIASH71.PROD.COBOL(CALTRAC)                            00040000
//* USING      VIASH71.PROD.COBOL($MACLIB$)                          00050000
//CALTRAC EXEC PGM=IGYCRCTL, REGION=2M,                               00060000
// PARM=('SIZE(MAX)', 'OPT', 'NORES', 'NODYN', 'NOADV', 'OBJ', 'APOST', 'LIB', 00070000
//   'MAP', 'NONUM', 'OFF', 'NOOBJ')                                  00080000
//*STEPLIB DD DSN=COB2.V402.COB2COMP, DISP=SHR                       00090000
//*SYSLIB DD DSN=VIASH71.PROD.COPYLIB, DISP=SHR                      00100000
//SYSPRINT DD SYSOUT=*                                               00110000
//SYSUT1 DD UNIT=SYSDA, SPACE=(CYL,(1,2)),                          00120000
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=32720)                          00130000
//SYSPUNCH DD DUMMY                                                 00140000
//SYSTEM DD SYSOUT=*                                                00150000
//SYSIN DD DISP=SHR, DSN=VIASH71.PROD.COBOL(CALTRAC)                00160000
//SYSOUT DD UNIT=SYSDA, SPACE=(CYL,(1,2))                            00170000

```


## Scan and submit JCL

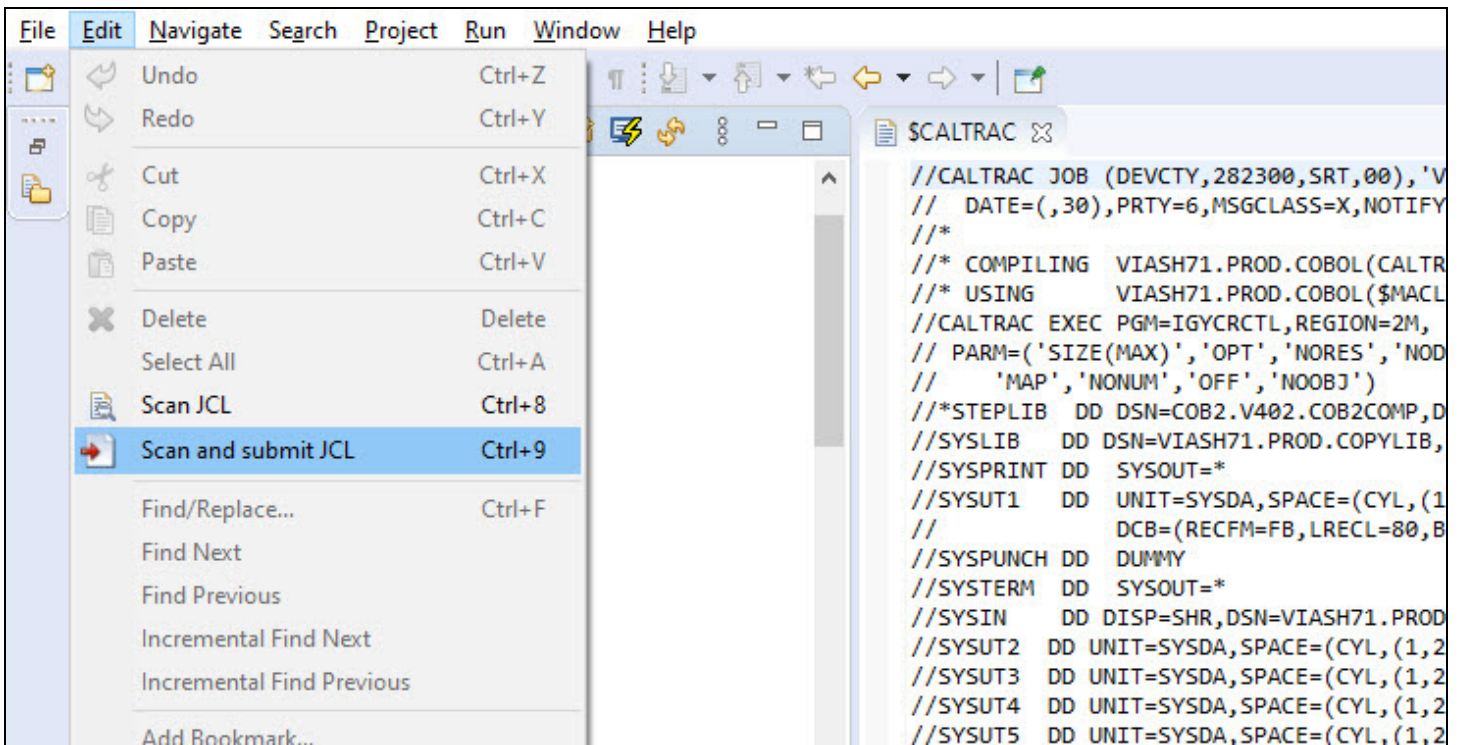
Scan and submit JCL option available in the ASG-JOB/SCAN Plug-in is similar to the ASG-JOB/SCAN's JSUB edit macro. The JJSUB edit macro validates a JCL and submits the job from the edit session. Similarly, the 'Scan and submit JCL' option validates a file that is active in the JCL Editor and submits the job to z/OS. If the validation process completes with a severity level of 4 or less, the job is automatically submitted and no reports are displayed in the JCL Scan or Problems views. If the severity level is greater than 4, then the job is not submitted and the reports are displayed.



- If you are not connected to a JOB/SCAN server when you run the JOB/SCAN commands, you will be asked to connect to JOB/SCAN first before the commands are executed. Perform the steps in [Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator](#) to connect to JOB/SCAN's server.
- When a file is opened from **Remote Systems** or **Project Explorer** view, JOB/SCAN commands are available in JCL Editor **only**.
- JCL scan options and processing options must be specified before you begin to scan a JCL file. For more information on setting these options, see [Setting the JCL Scan Options](#) and [Setting the Processing Options](#).

### ***To scan and submit JCL:***

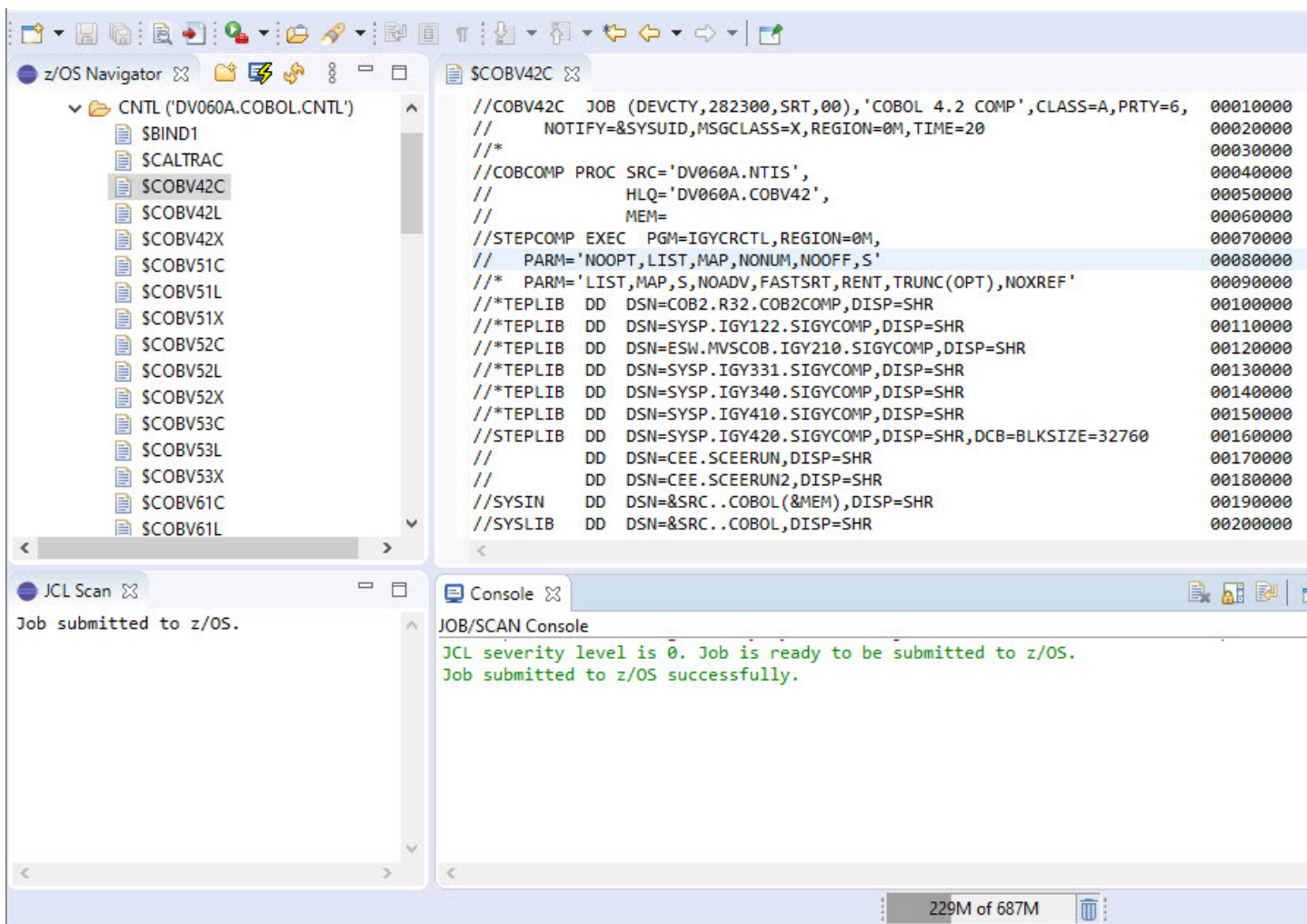
1. Open the file that you want to scan in JCL Editor.
2. Make changes, if any, and save it.
3. Select **Scan and submit JCL** option from the **Edit** menu or  icon from the toolbar.



4. JOB/SCAN Plug-in validates the input JCL and displays the reports in the JCL Scan and Problems views depending on the severity. The job will be submitted if the severity is less than or equal to 4.

#### Sample JCL file with severity level less than or equal to 4

The below sample JCL file with severity level less than or equal to 4 is automatically submitted to z/OS and a success message is displayed in the JCL Scan and JOB/SCAN Console views.



#### Sample JCL file with severity level greater than 4

The below sample JCL file with severity level greater than 4 is not submitted and a failure message along with severity level is displayed in the JOB/SCAN Console view. Validation reports are displayed in JCL Scan and Problems views. For more information on reports, refer to [Reports](#).

The screenshot displays the z/OS Navigator interface with four main panels:

- z/OS Navigator:** Shows a tree view of the file system under 'CNTL (DV060A.COBOL.CNTL)', listing files like SBIND1, SCALTRAC, and various SCOBV files.
- SCALTRAC:** Displays the JCL code for the SCALTRAC job, including parameters like DATE, PRTY, MSGCLASS, NOTIFY, and various DD statements for compilation and execution.
- Problems:** Shows a summary of 2 errors and 0 warnings. It lists 'Errors (2 items)' and 'Infos (1 item)'.
- JCL Scan:** Shows the scan results for the JCL code, highlighting an error: '\*\*\*ERROR - DSS1080E - INVALID JOB PARAMETER - "DATE "'.
- Console:** Shows the output of the JCL execution, including login messages and a final message: 'JCL severity level is 8. Job is not ready to be submitted to z/OS.'



When you scan another file in the JCL editor, the report in the JCL Scan view is replaced. However, the messages in the Problems view are not replaced, instead they add up to the existing messages.

# Reports

ASG-JOB/SCAN Plug-in validates the input JCL for any errors, warnings and advisories (informational). The results of the validation are displayed in an easy-to-read report format. These reports are displayed automatically to the user in the **JCL Scan** and **Problems** views.

The reports contain the following messages:

## Error

These messages are likely to affect processing and should be corrected before the JCL is submitted.

## Warning

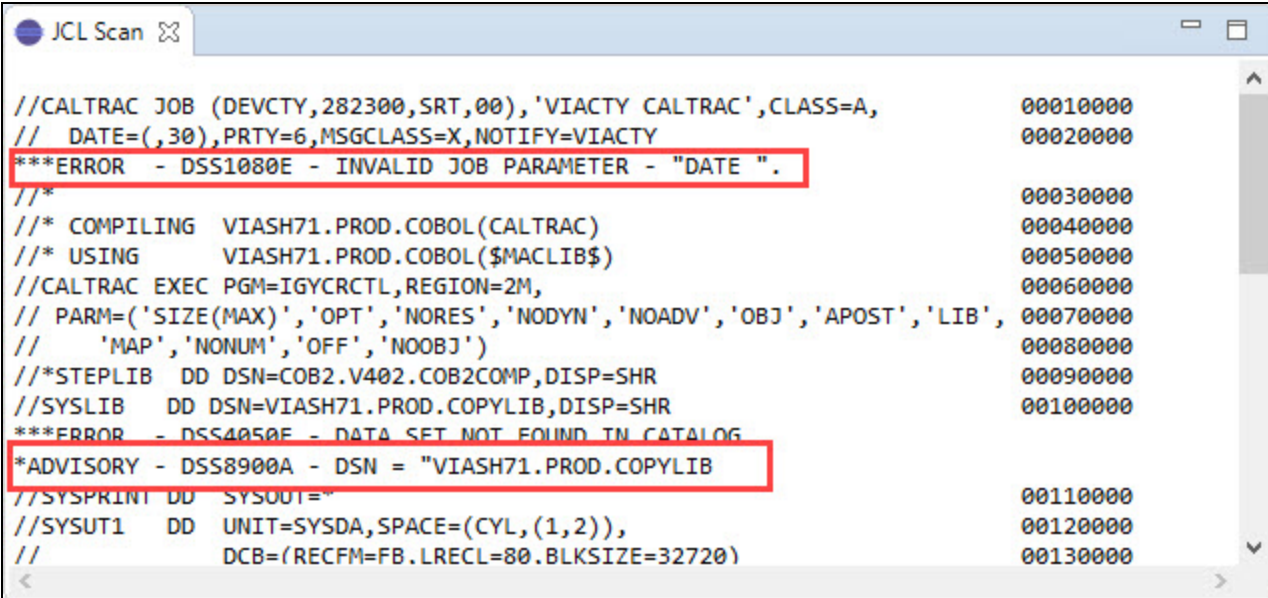
These messages may or may not affect processing and should be examined to determine if they must be corrected before the JCL is submitted.

## Advisory

These messages are informational only and are unlikely to affect processing.

## JCL Scan Report

In the JCL Scan report, the messages appear immediately after the JCL line that contains errors or warnings or advisories.



```

JCL Scan
//CALTRAC JOB (DEVCTY,282300,SRT,00),'VIACTY CALTRAC',CLASS=A,          00010000
// DATE=(,30),PRTY=6,MSGCLASS=X,NOTIFY=VIACTY                        00020000
***ERROR - DSS1080E - INVALID JOB PARAMETER - "DATE ".
//*                                                                    00030000
//* COMPILING VIASH71.PROD.COBOBOL(CALTRAC)                          00040000
//* USING      VIASH71.PROD.COBOBOL($MACLIB$)                        00050000
//CALTRAC EXEC PGM=IGYCRCTL,REGION=2M,                                00060000
// PARM=('SIZE(MAX)', 'OPT', 'NORES', 'NODYN', 'NOADV', 'OBJ', 'APOST', 'LIB', 00070000
//   'MAP', 'NONUM', 'OFF', 'NOOBJ')                                00080000
//*STEPLIB DD DSN=COB2.V402.COBOB2COMP,DISP=SHR                      00090000
//SYSLIB DD DSN=VIASH71.PROD.COPYLIB,DISP=SHR                       00100000
***ERROR - DSS4050E - DATA SET NOT FOUND IN CATALOG
*ADVISORY - DSS8900A - DSN = "VIASH71.PROD.COPYLIB
//SYSPRINT DD SYSOUT=*                                             00110000
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,2)),                          00120000
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=32720)                          00130000

```

The messages are displayed in the following format:

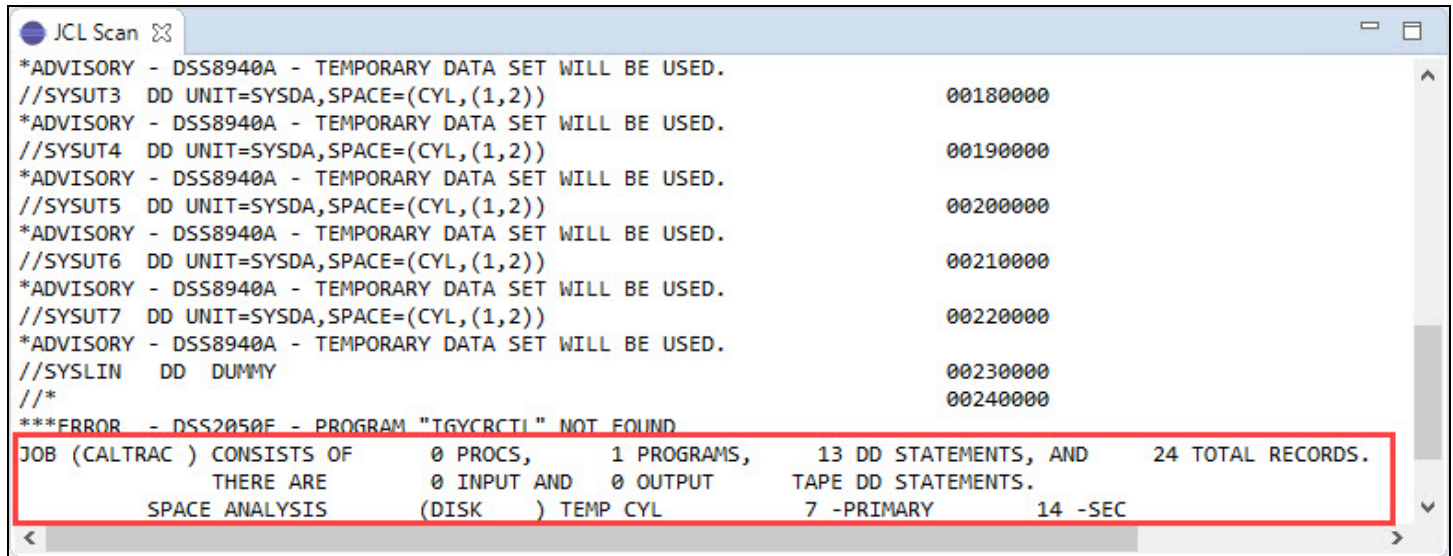
[Message Type] - [Message Number] - [Message Text]

- Message types are:
  - Advisory
  - Error

- Warning
- Message number is a five digit numeral prefixed with DSS and suffixed with message type.
- Message text is a brief description of the problem.

After the messages, the report summarizes the Statistical Summary and the Space Analysis.

The Statistical Summary includes the total number of PROGRAM(S), PROC(S), INCLUDE(S), DD STATEMENT(S) and input / output TAPE DD STATEMENT(S) that are in the scanned JCL file.



```

JCL Scan
*ADVISORY - DSS8940A - TEMPORARY DATA SET WILL BE USED.
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,2))                00180000
*ADVISORY - DSS8940A - TEMPORARY DATA SET WILL BE USED.
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(1,2))                00190000
*ADVISORY - DSS8940A - TEMPORARY DATA SET WILL BE USED.
//SYSUT5 DD UNIT=SYSDA,SPACE=(CYL,(1,2))                00200000
*ADVISORY - DSS8940A - TEMPORARY DATA SET WILL BE USED.
//SYSUT6 DD UNIT=SYSDA,SPACE=(CYL,(1,2))                00210000
*ADVISORY - DSS8940A - TEMPORARY DATA SET WILL BE USED.
//SYSUT7 DD UNIT=SYSDA,SPACE=(CYL,(1,2))                00220000
*ADVISORY - DSS8940A - TEMPORARY DATA SET WILL BE USED.
//SYSLIN DD DUMMY                                       00230000
//*                                                       00240000
***ERROR - DSS2050E - PROGRAM "TGYCRCTI" NOT FOUND
JOB (CALTRAC ) CONSISTS OF      0 PROCS,      1 PROGRAMS,      13 DD STATEMENTS, AND      24 TOTAL RECORDS.
      THERE ARE      0 INPUT AND      0 OUTPUT      TAPE DD STATEMENTS.
      SPACE ANALYSIS      (DISK ) TEMP CYL      7 -PRIMARY      14 -SEC
  
```

The Space Analysis includes list of all new space requirements for the validated job stream. The total allocation requirements are broken down by UNIT and VOLSER, and indicate whether a file is permanent or temporary.

Next, the ERROR SUMMARY AND COUNTS section summarizes the total number of errors, warnings, and advisories encountered after scanning the JCL. This section is displayed only when you select the **Include summary in JCL scan output** option in the **JCL Scan Options** tab in the **JOB/SCAN** preference page.

```

JCL Scan
*** ERROR SUMMARY AND COUNTS ***
 9 ADVISORY LEVEL
 0 WARNING LEVEL
 3 ERROR LEVEL
12 TOTAL ISSUED      0 SUPPRESSED.
  THE HIGHEST SEVERITY CODE ENCOUNTERED WAS      8

Options in Effect
-----
RUN OPTIONS
  JLJPC PROCLIB TYPE=PDS
  USE ERROR TABLE =NO
  PROCESSING SPECIFICATIONS =NOCONDCODE
  CATALOG ACCESS =YES,VTOC,PDSMEM,PGM
  PROGRAM PRODUCT INTERFACE =IDCAMS
  SCHEDULER =
  ACCESS CONTROL CARDS =YES,LIMIT=0
  SUBSYSTEM =NONE
  JOB LIMIT =0
  SUPPRESS ERROR MESSAGES =NO
REPORT OPTIONS
  STRUCTURED JCL LISTING=PRINT,ERRORTXT=SHORT
INPUT SELECTION
  DDNAME=JCLIN
  SOURCE TYPE=SEQ
FILE ALLOCATIONS
  JLJPC=SHR,DSN=DV405A.JCL

```

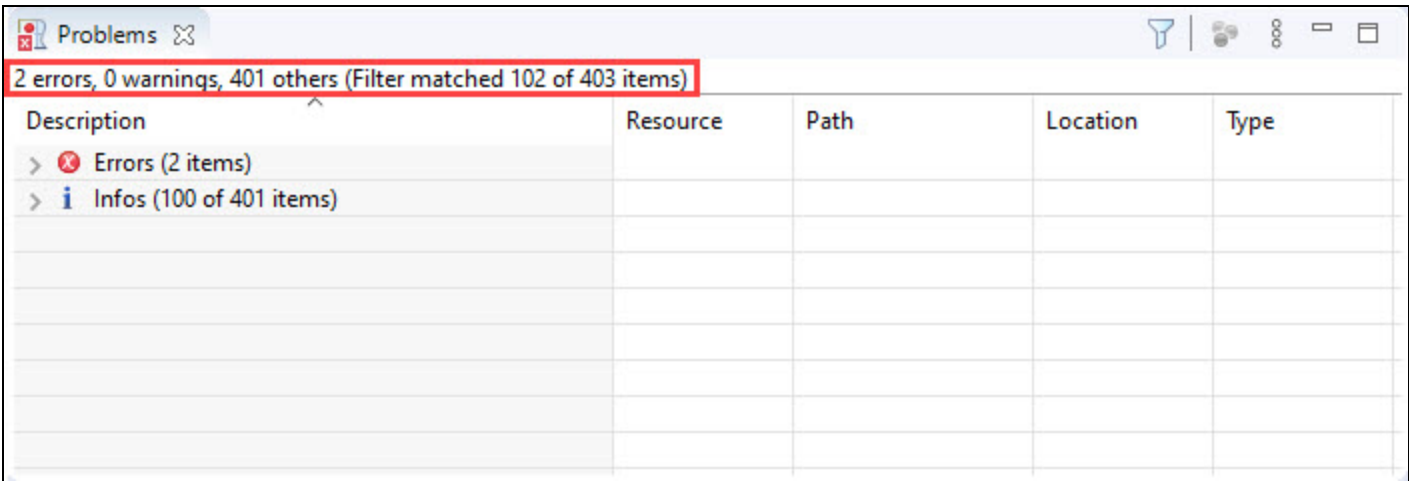
At the end is the Options in Effect section. This section summarizes the JCL scan options and the processing options that are used in validating a JCL. This section is displayed only when you select the **Include summary in JCL scan output** option in the **JCL Scan Options** tab in the **JOB/SCAN** preference page.

## Problems Report

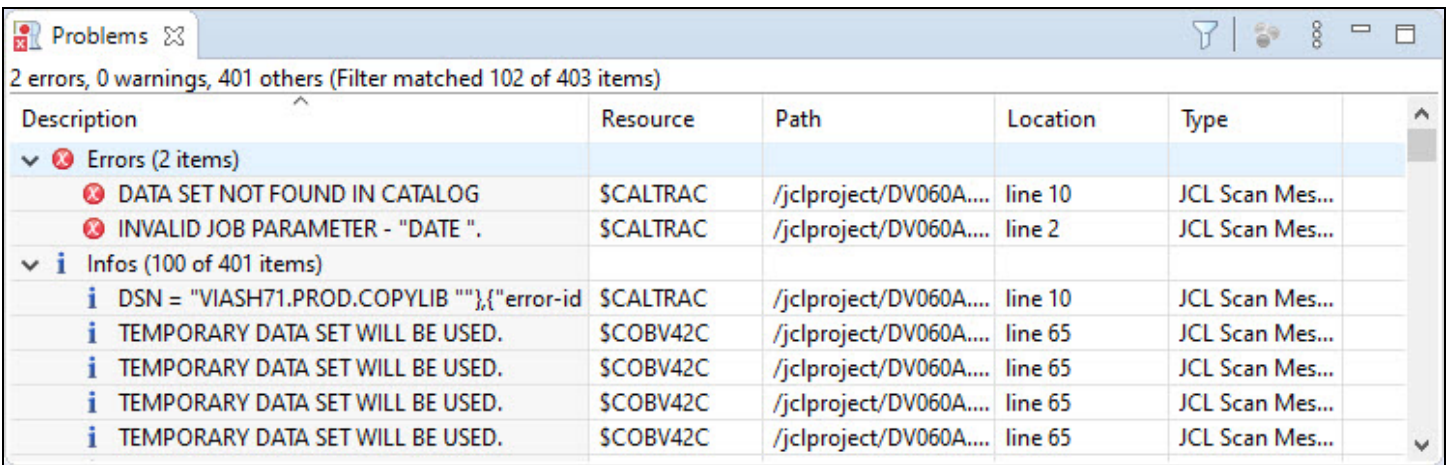
The Problems report displays the list of messages that are identified by the ASG-JOB/SCAN Plug-in.

At the top, the report summarizes the number of errors, warnings, and advisories encountered. Subsequent section displays the list of messages in each category.





Expand a category by clicking the > icon to view the list.



Information about the messages are listed in a table format containing the following columns.

Column	Description
--------	-------------

- Description** Message describing the problem
- Resource** Member name in PDS or name of sequential dataset
- Path** Partitioned dataset name or sequential dataset
- Location** Indicates the JCL line number where the error appears
- Type** *JCL Scan Message*

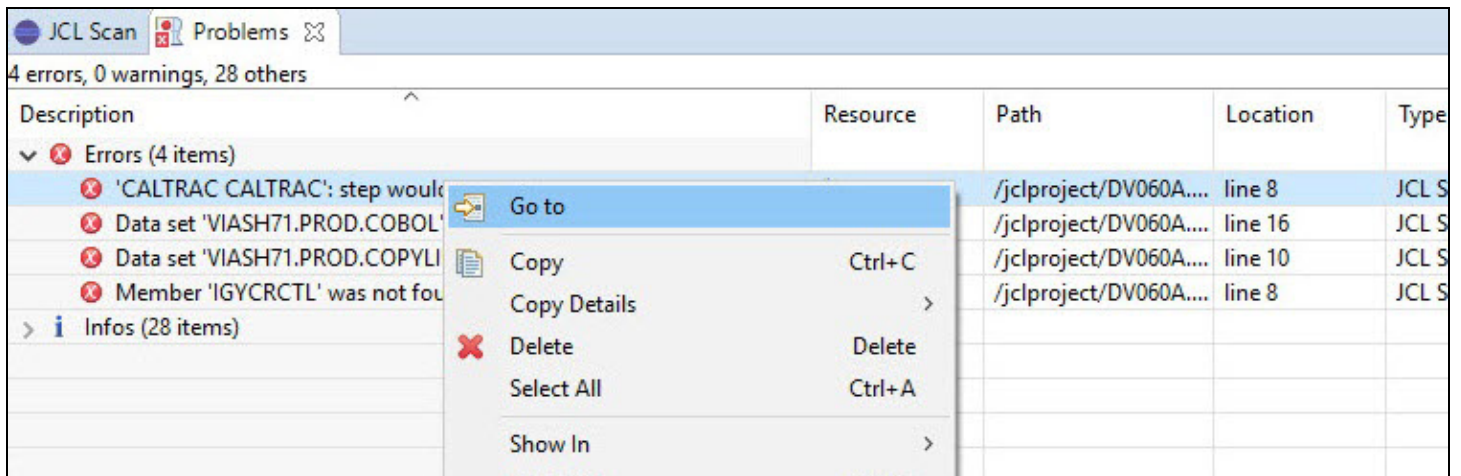
JOB/SCAN's JJFE edit macro locates a specific error in the Structured JCL Listing (SJL) output, similarly the Problems view offers a simple and quick way to locate the problem line that is associated with the message.

You can locate the problem either by:

- double-clicking on the message

Or

- right-click on the message to bring up the problems view's contextual menu and select the **Go to** option.



# Markers

Markers are small indicators that appear on the right and left margins of the JCL editor after a file is validated. These indicators enable you to quickly identify problems in your file. With these markers, user can navigate between the problem lines and identify the highest severity in a file.

Markers are categorized as follows:

- Errors
- Warnings
- Advisories (Information)



By default, only the error markers are displayed in the right and left margins. You can choose to show / hide markers as per your preference. See [Show or Hide Markers](#) for more information.

## Left-margin Markers

Markers on the left-margin are placed next to the lines that are associated with some problems. These markers move with the source lines when the content in the JCL editor is scrolled.

The left-margin can have different icons with the following meaning:

Icon	Description
	Errors
	Warnings
	Advisories

```

XMIT2AKR
//XMIT2AKR JOB (DEVCTY,282300,SRT,00), 'DUMP XMIT',CLASS=A,MSGCLASS=X, 00010000
//          NOTIFY=&SYSUID,TIME=(4,30),PRTY=6,REGION=4096K           00020000
//          *                                                         00030000
//DUMP      EXEC PGM=DUMPXMIT,PARM='LOGICAL,TERSE'                   00040000
//STEPLIB  DD  DSN=DV060A.DEVL.LOAD,DISP=SHR                          00050000
//SYSIN    DD  DSN=CUSTDUMP.T768917.XMI,DISP=SHR                      00060000
//SYSPRINT DD  SYSOUT=*                                               00070000
//SYSOUT   DD  DSN=DV060A.HUGE.AKR,DISP=SHR                           00080000
//*SYSOUT  DD  DSN=DV060A.TEST.HEXDUMP.DUMPXMIT,DISP=(NEW,CATLG),    00090000
//          UNIT=SRTDA,SPACE=(CYL,(1,1))                             00100000
//          *
//          *

```

When you hover the mouse cursor over a marker, a pop-up shows the description of the problem(s) associated with the line.

```

XMIT2AKR
//XMIT2AKR JOB (DEVCTY,282300,SRT,00), 'DUMP XMIT',CLASS=A,MSGCLASS=X, 00010000
//      NOTIFY=&SYSUID,TIME=(4,30),PRTY=6,REGION=4096K 00020000
//* 00030000
//DUMP EXEC PGM=DUMPXMIT,PARM=' LOGICAL,TERSE' 00040000
//STEPLIB DD DSN=DV060A.DEVL.LOAD,DISP=SHR 00050000
Data set 'CUSTDUMP.T768917.XMI' was not found in catalog. 00060000
//SYSPRINT DD SYSOUT=* 00070000
//SYSOUT DD DSN=DV060A.HUGE.AKR,DISP=SHR 00080000
//*YSOUT DD DSN=DV060A.TEST.HEXDUMP.DUMPXMIT,DISP=(NEW,CATLG), 00090000
//* UNIT=SRTDA,SPACE=(CYL,(1,1)) 00100000
//*

```

## Right-margin Markers

Markers on the right-margin show the relative locations of the tagged lines in the file. As a result, the markers are still visible when the margin shrinks to fit the screen. These markers do not move when the content in the editor is scrolled. Instead, clicking any marker in the right-margin would highlight the line associated with that marker. This allows you to quickly step through the problem lines in the editor by simply clicking the markers one by one.

Markers on the right-margin are color coded according to the message type:

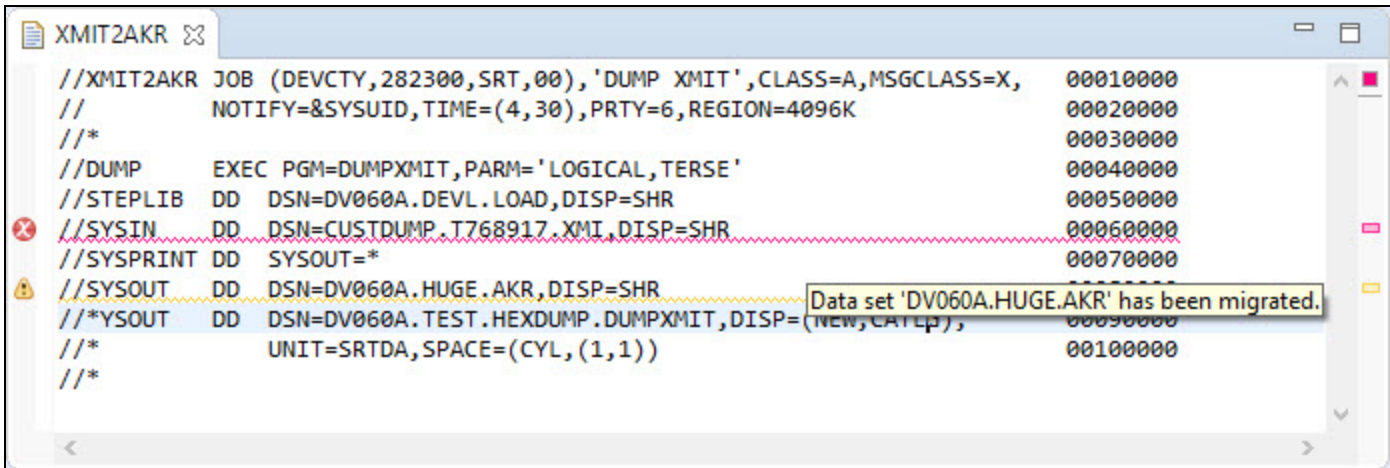
Color	Description
Blue	Advisory messages
Yellow	Warning messages
Red	Error messages

```

XMIT2AKR
//XMIT2AKR JOB (DEVCTY,282300,SRT,00), 'DUMP XMIT',CLASS=A,MSGCLASS=X, 00010000
//      NOTIFY=&SYSUID,TIME=(4,30),PRTY=6,REGION=4096K 00020000
//* 00030000
//DUMP EXEC PGM=DUMPXMIT,PARM=' LOGICAL,TERSE' 00040000
//STEPLIB DD DSN=DV060A.DEVL.LOAD,DISP=SHR 00050000
Data set 'CUSTDUMP.T768917.XMI' was not found in catalog. 00060000
//SYSPRINT DD SYSOUT=* 00070000
//SYSOUT DD DSN=DV060A.HUGE.AKR,DISP=SHR 00080000
//*YSOUT DD DSN=DV060A.TEST.HEXDUMP.DUMPXMIT,DISP=(NEW,CATLG), 00090000
//* UNIT=SRTDA,SPACE=(CYL,(1,1)) 00100000
//*

```

When you hover the mouse cursor over a marker, a pop-up shows the description of the problem(s) associated with the line.



```

XMIT2AKR
//XMIT2AKR JOB (DEVCTY,282300,SRT,00),'DUMP XMIT',CLASS=A,MSGCLASS=X, 00010000
//      NOTIFY=&SYSUID,TIME=(4,30),PRTY=6,REGION=4096K 00020000
//* 00030000
//DUMP EXEC PGM=DUMPXMIT,PARM='LOGICAL,TERSE' 00040000
//STEPLIB DD DSN=DV060A.DEVL.LOAD,DISP=SHR 00050000
//SYSIN DD DSN=CUSTDUMP.T768917.XMI,DISP=SHR 00060000
//SYSPRINT DD SYSOUT=* 00070000
//SYSOUT DD DSN=DV060A.HUGE.AKR,DISP=SHR 00080000
//*YSOUT DD DSN=DV060A.TEST.HEXDUMP.DUMPXMIT,DISP=(NEW,CATLG), 00090000
//* UNIT=SRTDA,SPACE=(CYL,(1,1)) 00100000
//*

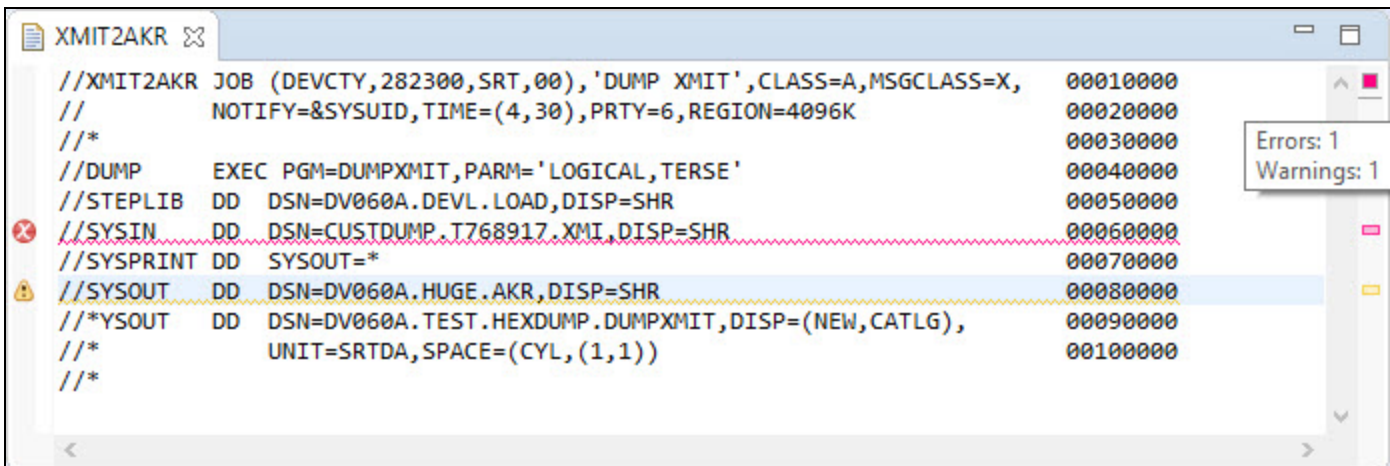
```

The square that appears at the top right-margin shows the highest severity found in the file. Messages have the following severity levels:

Severity	Color	Description
High	Red	Error messages
Medium	Yellow	Warning messages
Low	Blue	Advisory messages

The square is not displayed if there are no messages associated with the file.

When you hover the mouse cursor over the square, a pop-up shows the total number of severities encountered in the current file.



```

XMIT2AKR
//XMIT2AKR JOB (DEVCTY,282300,SRT,00),'DUMP XMIT',CLASS=A,MSGCLASS=X, 00010000
//      NOTIFY=&SYSUID,TIME=(4,30),PRTY=6,REGION=4096K 00020000
//* 00030000
//DUMP EXEC PGM=DUMPXMIT,PARM='LOGICAL,TERSE' 00040000
//STEPLIB DD DSN=DV060A.DEVL.LOAD,DISP=SHR 00050000
//SYSIN DD DSN=CUSTDUMP.T768917.XMI,DISP=SHR 00060000
//SYSPRINT DD SYSOUT=* 00070000
//SYSOUT DD DSN=DV060A.HUGE.AKR,DISP=SHR 00080000
//*YSOUT DD DSN=DV060A.TEST.HEXDUMP.DUMPXMIT,DISP=(NEW,CATLG), 00090000
//* UNIT=SRTDA,SPACE=(CYL,(1,1)) 00100000
//*

```

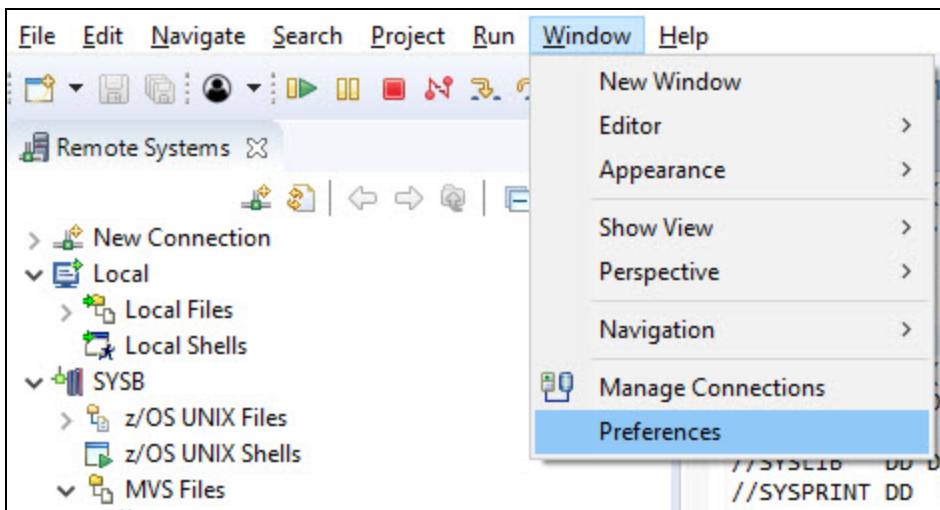
## Show or Hide Markers

Markers like errors, warnings and infos are used to quickly identify the problems in a file. You might find these markers useful when you are validating a JCL file. You may choose how you want these markers to be displayed through the

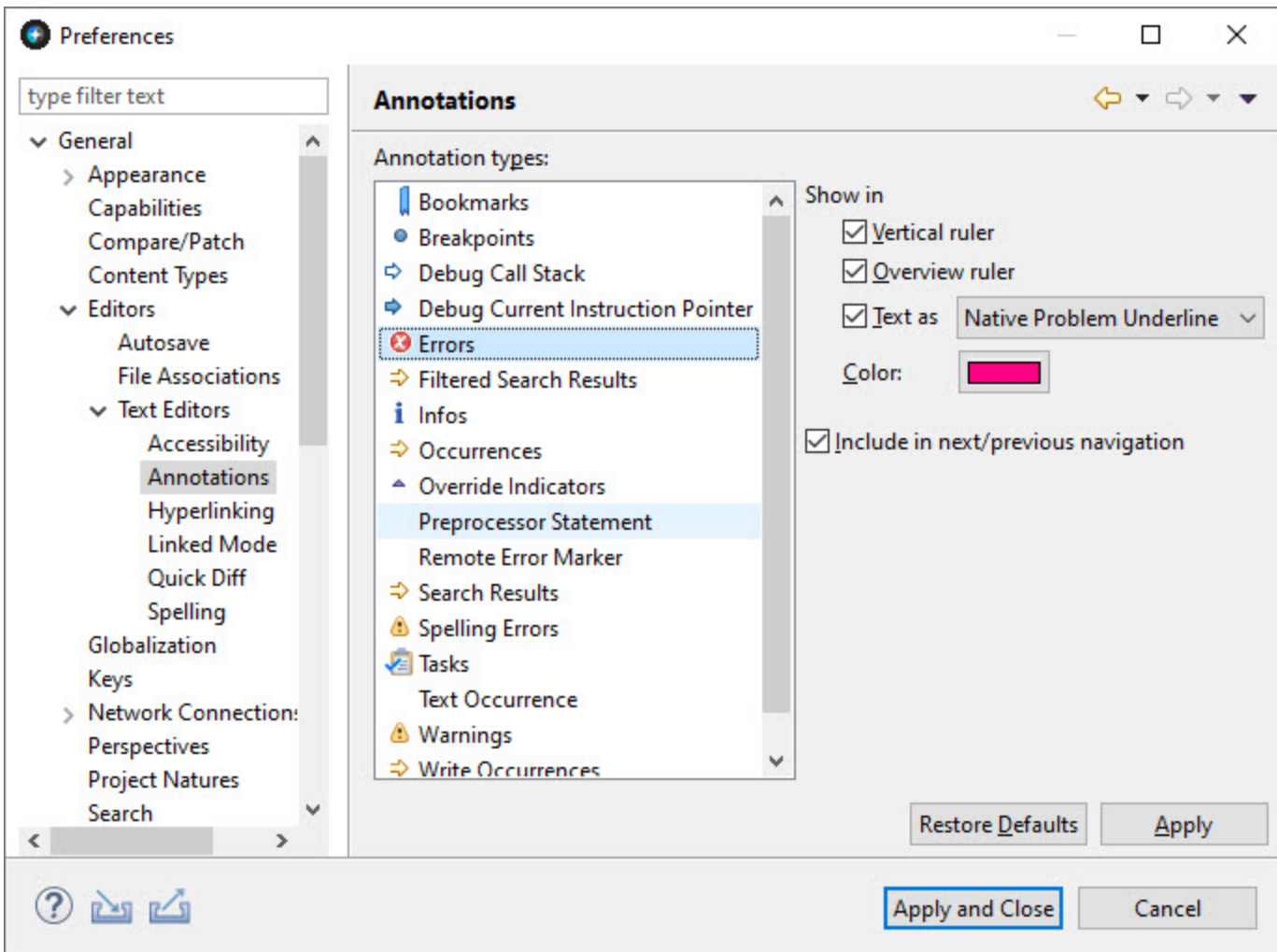
Preferences dialog.

**To show or hide markers:**

1. From the **Window** menu, select **Preferences**.



2. In the **Preferences** dialog, select **General > Editors > Text Editors > Annotations**.



3. In the **Annotations** page, under the **Annotations types**, select **Errors / Infos / Warnings** as per your requirement.
4. Under **Show in**, select or deselect the following options:

Options	Description
Vertical ruler	This option control the markers on the left-margin.
Overview ruler	This option control the markers on the right-margin.
Text as	This option control the format of the tagged lines in the JCL editor region. You can select on of the following formats: <ul style="list-style-type: none"> <li>• Box</li> <li>• Dashed Box</li> <li>• Highlighted</li> <li>• Native Problem Underline</li> </ul>

Options	Description
	<ul style="list-style-type: none"><li>• Squiggly Line</li><li>• Underlined</li><li>• Vertical Bar</li></ul>

5. After choosing your preference, click **Apply** to save the changes or click **Apply and Close** to save the changes and close the **Annotations** dialog.



# Keyboard Shortcuts

In this topic you can find the list of all the keyboard shortcuts for different tasks performed in ASG-JOB/SCAN Plug-in. Also, the topic explains the procedure to customize the default shortcuts.

The following table shows the shortcut keys used in ASG-JOB/SCAN Plug-in:

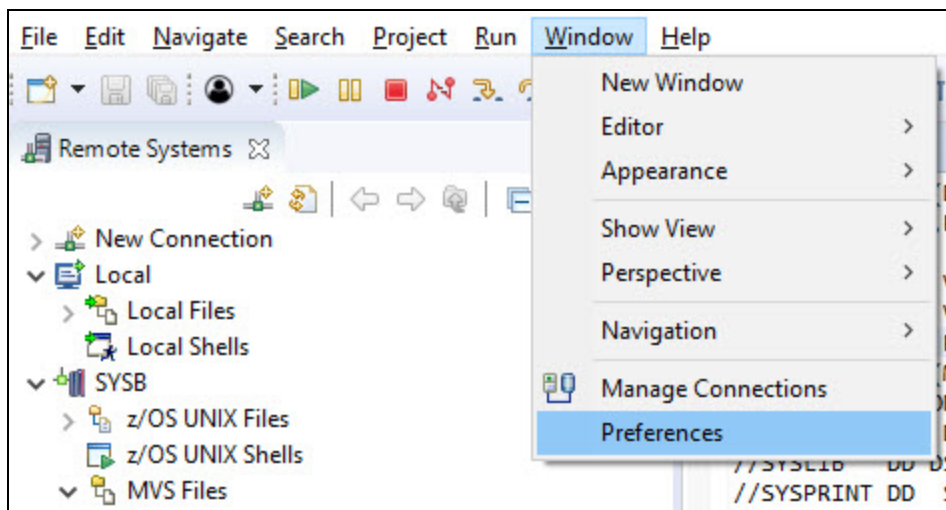
Task	Default Shortcut
Scan JCL	Ctrl + 8
Scan and submit JCL	Ctrl + 9
Save to z/OS	Ctrl + 5
Save to z/OS As...	Ctrl + Shift + 5

## Customizing Keyboard Shortcuts

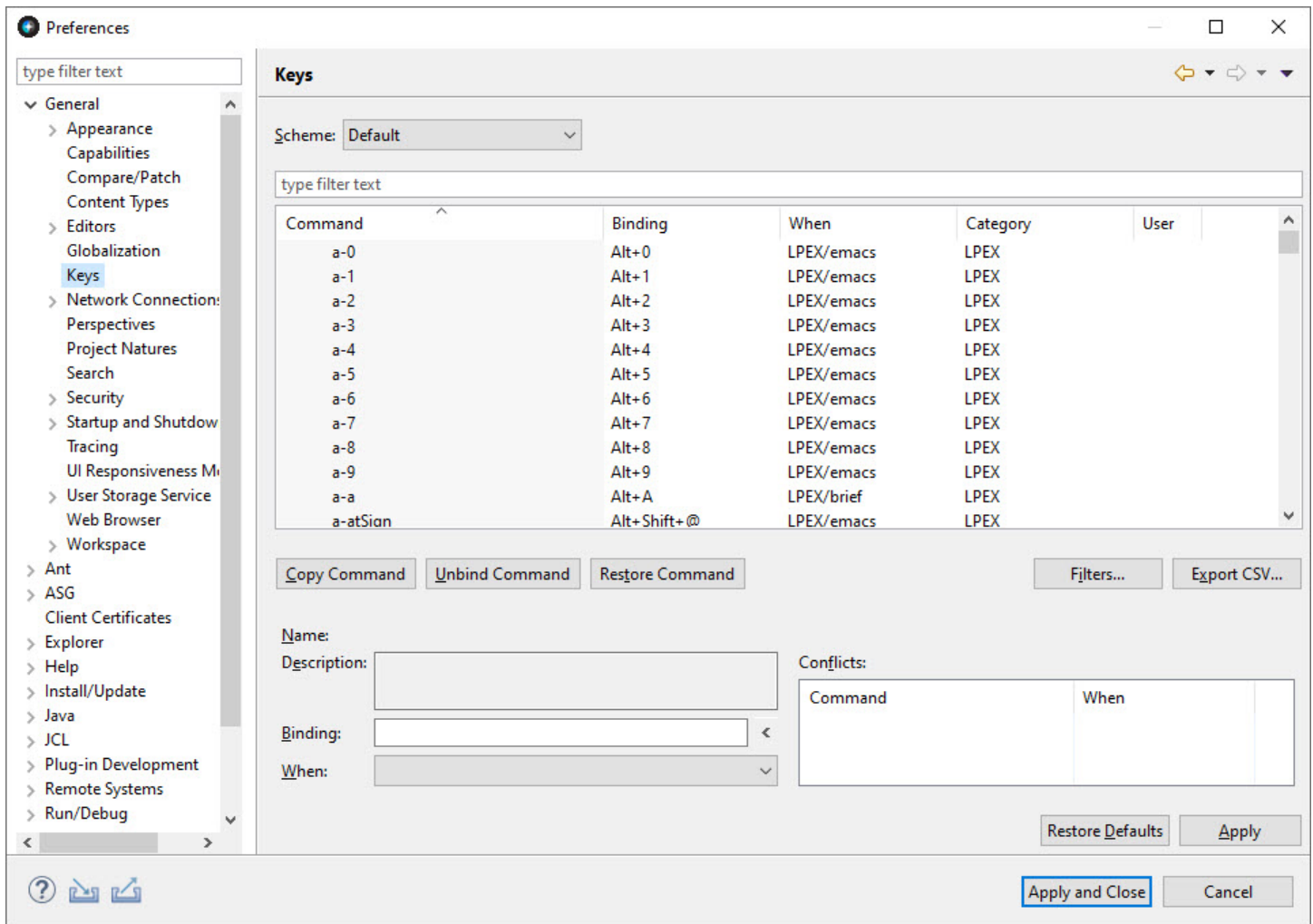
The key combinations for the existing shortcuts can be disabled or modified as required.

**To customize shortcuts:**

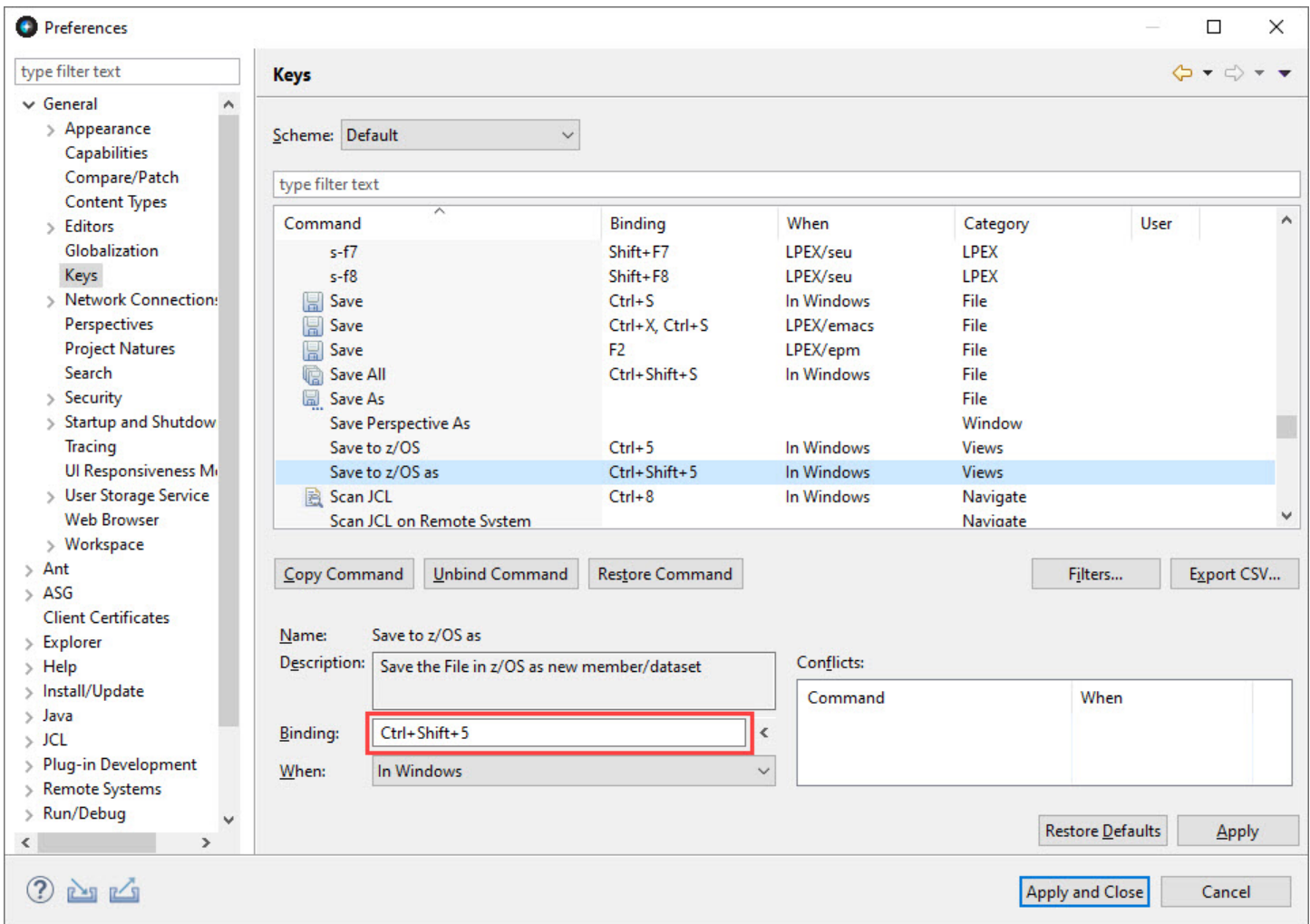
1. From the **Window** menu, select **Preferences**.



2. In the **Preferences** dialog, select **General > Keys**.



- In the **Keys** page, select the **Command** that you like to change. The values associated with the command is populated in the fields.



4. To change the key combinations, edit the **Binding** field.
5. If required, you can edit the values in the other fields.
6. click **Apply** to save the changes or click **Apply and Close** to save the changes and close the **Preferences** dialog.

# ASG-JOB/SCAN REST API Documentation

The ASG-JOB/SCAN REST API service is used to interact with ASG-JOB/SCAN REST server. It gives the ability to scan the JCLs from any application. The ASG-JOB/SCAN REST API service is defined using the OpenAPI 3.0 specification.

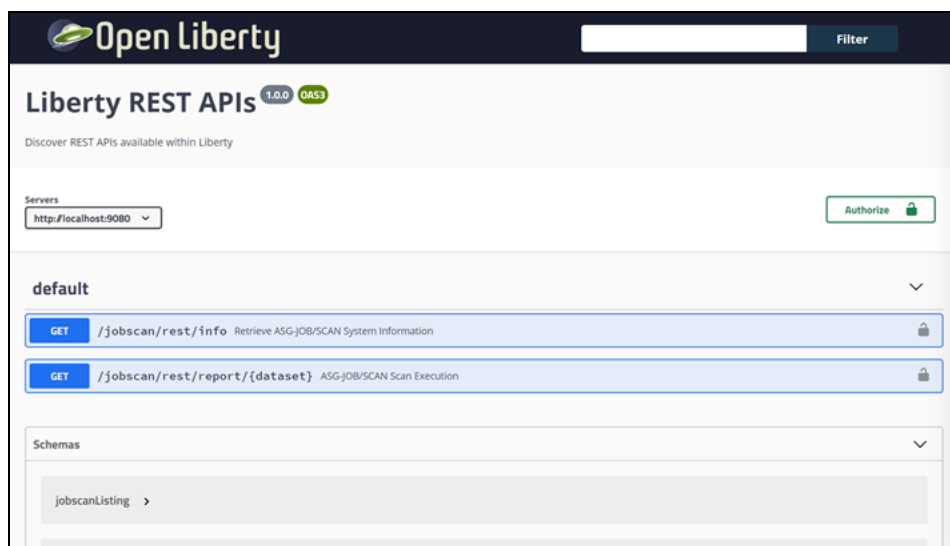
The ASG-JOB/SCAN REST API user interface is accessible at the URL: `http://<Host_Name>:<Port>/api/explorer/`

wherein:

*Host\_Name* indicates the host name or IP address of the z/OS system

*Port* indicates the port number of the host machine on which the JOB/SCAN REST API service is running

Example: `http://mvssyse.asg.com:60080/api/explorer/`

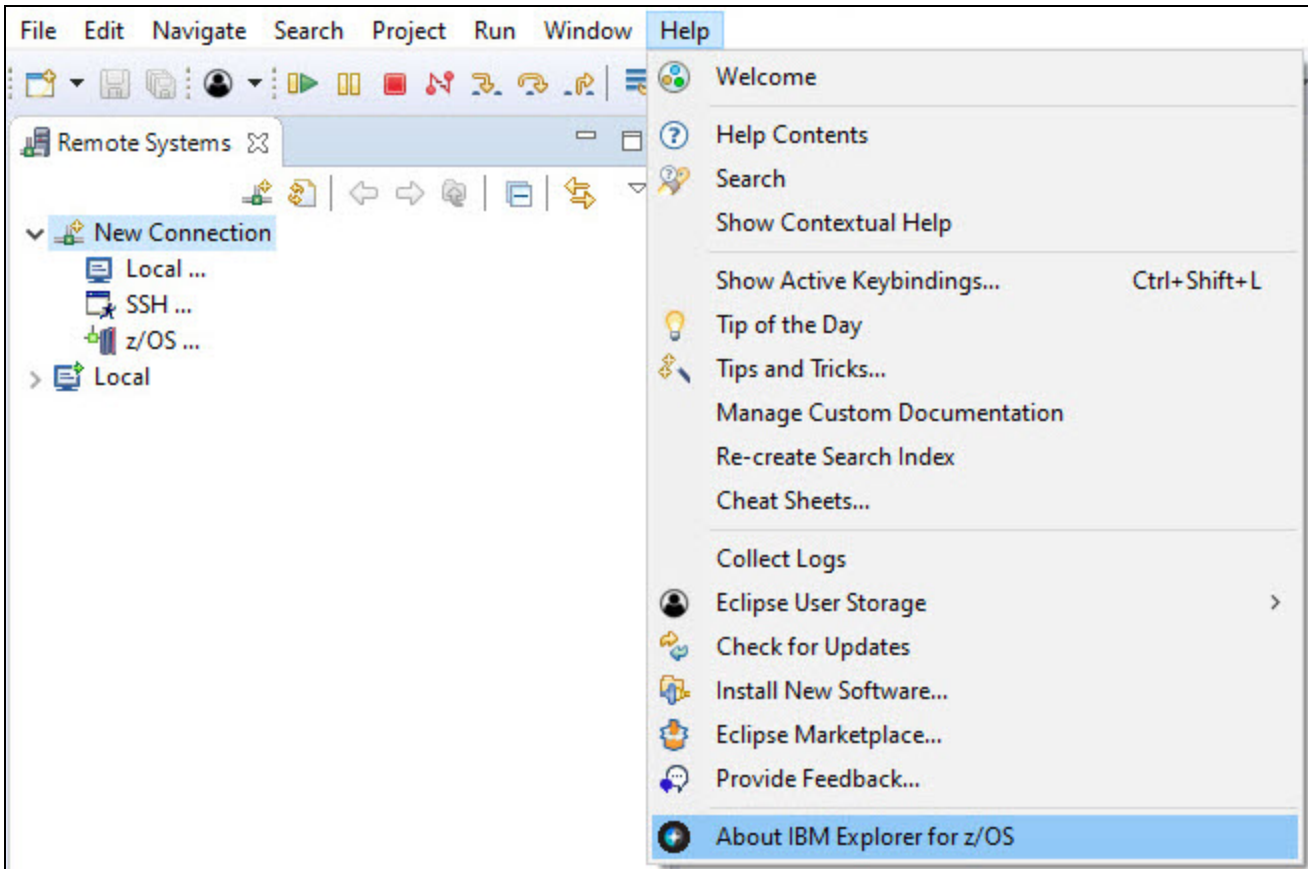


# Uninstallation

Before installing the new version of ASG-JOB/SCAN Plug-in, it is recommended to remove any prior versions from the server.

## To uninstall ASG-JOB/SCAN Plug-in:

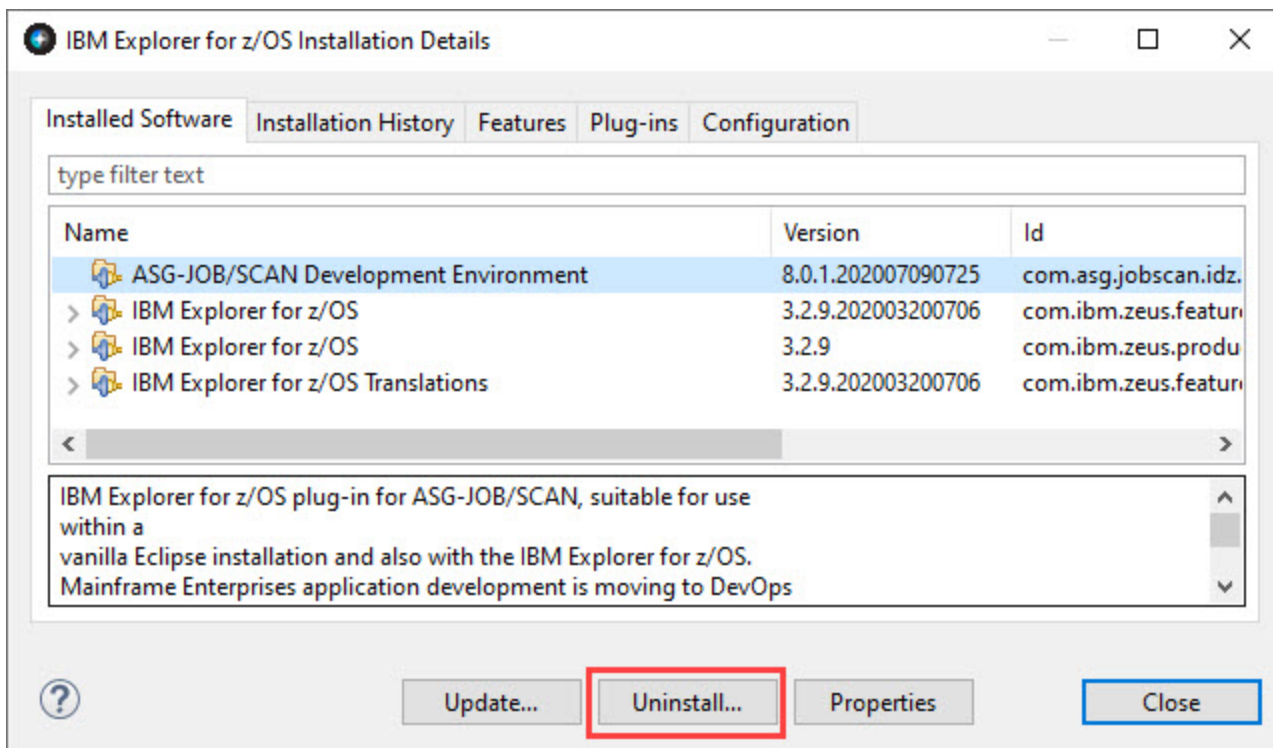
1. Start z/OS Explorer.
2. From the menu, select **Help > IBM Explorer for z/OS**.



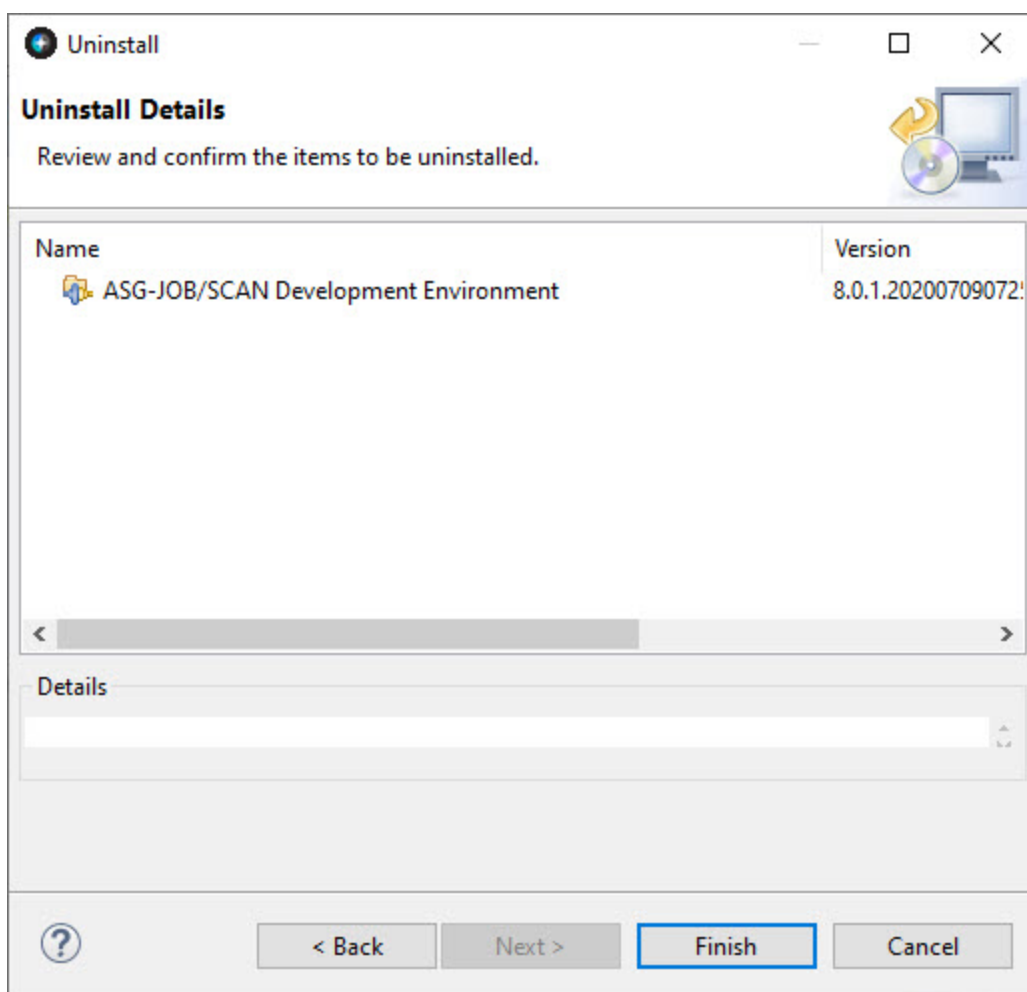
3. In the **About IBM Explorer for z/OS** dialog, click **Installation Details**.



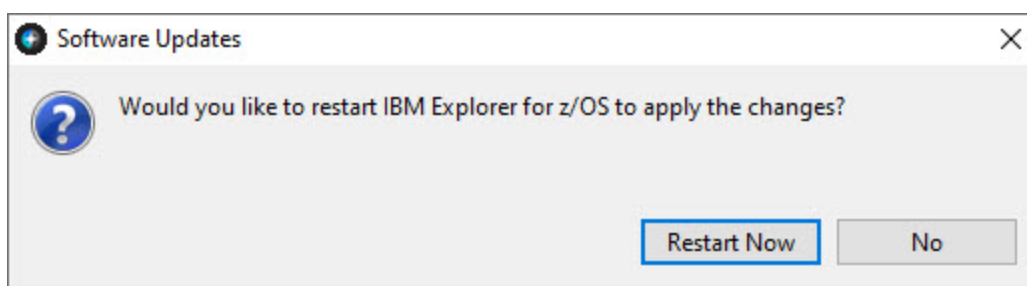
- In the **Installed Software** tab, select **ASG-JOB/SCAN Development Environment** and click **Uninstall...**



- Review the uninstall items and click **Finish**.



6. When the uninstallation is complete, you are prompted to restart z/OS Explorer. Click **Restart Now**.



**To verify if the plug-in has been uninstalled:**

1. Start z/OS Explorer.
2. From the menu, select **Help > About IBM Explorer for z/OS**.
3. In the **About IBM Explorer for z/OS** dialog, click **Installation Details**.
4. In the **Installed Software** tab, you should not see **ASG-JOB/SCAN Development Environment** in the list of installed software. This indicates that ASG-JOB/SCAN Plug-in has been uninstalled.

# ASG-JOB/SCAN Plug-in for Topaz Workbench

This documentation introduces the features and benefits of ASG-JOB/SCAN Plug-in, explains how to install, and execute the product in Compuware Topaz Workbench.

It consist of following topics:

- [Introduction](#)
- [Installation](#)
- [User Interface](#)
- [Connecting to z/OS from Host Explorer](#)
- [Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator](#)
- [Setting the JCL Scan Options](#)
- [Setting the Processing Options](#)
- [Working with z/OS Datasets](#)
- [Working with Local Files](#)
- [Validating JCL](#)
- [Reports](#)
- [Markers](#)
- [Keyboard Shortcuts](#)
- [ASG-JOB/SCAN REST API Documentation](#)
- [Uninstallation](#)



# Introduction

## What is ASG-JOB/SCAN Plug-in?

ASG-JOB/SCAN Plug-in is a Java plug-in that strategically integrates the mainframe capabilities of ASG-JOB/SCAN into the Topaz Workbench. The Plug-in provides developers and production control staff an intuitive user interface to initiate JCL integrity, verification, and reporting. The Plug-in supports the key functions implemented in ASG-JOB/SCAN.



The JOB/SCAN Plug-in is a separately licensed option of ASG-JOB/SCAN. You need a license key to access it. Contact [ASG Customer Support](#) to obtain a license.

## Features of ASG-JOB/SCAN Plug-in:

- Offers a modern and easy-to-use User Interface (UI) to interact with the z/OS system.
- Evaluates JCL for any syntax errors.
- Enforces site-specific JCL standards.
- Submits jobs to the mainframe for execution.
- Provides effective reporting facility.
- Includes RESTful web services API to automate ASG-JOB/SCAN scans in a DevOps toolchain environment.

This document assumes that the user:

- has installed ASG-JOB/SCAN and running on a supported z/OS system.
- has authorization to access resources on the z/OS system.
- is familiar with the functions implemented in ASG-JOB/SCAN. If you are new to ASG-JOB/SCAN, refer to [ASG-JOB/SCAN® 8.0.0 Documentation](#).

# Installation

This topic contains instructions for installing ASG-JOB/SCAN Plug-in. These instructions include product requirements, installing the product, and verifying the installation.

## Installation Requirements

Before you install JOB/SCAN Plug-in, verify that the following requirements are met:

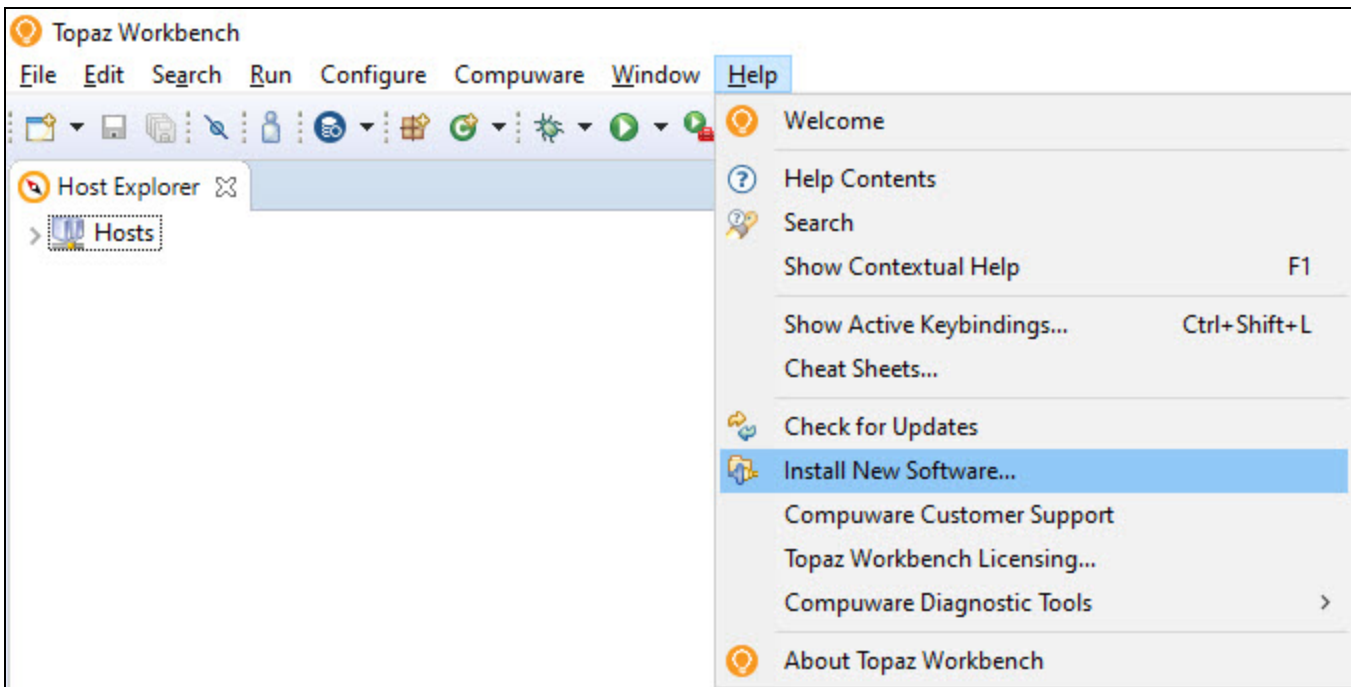
- ASG-JOB/SCAN Version 8.0.0 must be installed in the z/OS system. This topic does not explain the installation procedure for ASG-JOB/SCAN. See [ASG-JOB/SCAN Installation](#) for information on installing ASG-JOB/SCAN.
- Latest monthly cumulative service for ASG-JOB/SCAN must be applied. Visit ACCESS Portal: <https://access.asg.com> to verify any monthly maintenance apply to this product release.
- Install and Configure IBM Liberty Profile Server to enable ASG-JOB/SCAN REST Services. For details on installing Liberty server see the [Installer Reference Guide](#) section for guides on installing the ASG-JOB/SCAN REST Services. For more information about Liberty Servers z/OS installation requirements see [IBM Installing Liberty](#).
- User must have a TSO user ID defined to the mainframe system and should be able to access z/OS Unix System Services with home directory setup.
- User's TSO ID must be able to access and use ASG-JOB/SCAN under ISPF.
- Compuware Topaz Workbench version 20.01.02 or higher must be installed.
- Java Runtime Environment (JRE) must be installed.
  - You can install JRE from OpenJ9: <https://www.eclipse.org/openj9/>
  - You can install JRE from Oracle: <https://www.java.com/en/download/manual.jsp>
- IBM z/OS® Management Facility (z/OSMF) Version 2.4 must be installed and running before using ASG-JOB/SCAN Plug-in. User must have read and write access to z/OSMF. To obtain read or write access, contact your system administrator. To verify if z/OSMF is configured as per ASG-JOB/SCAN Plug-in requirements, follow the procedure in [Verifying the z/OSMF Configuration](#).

## Installing ASG-JOB/SCAN Plug-in

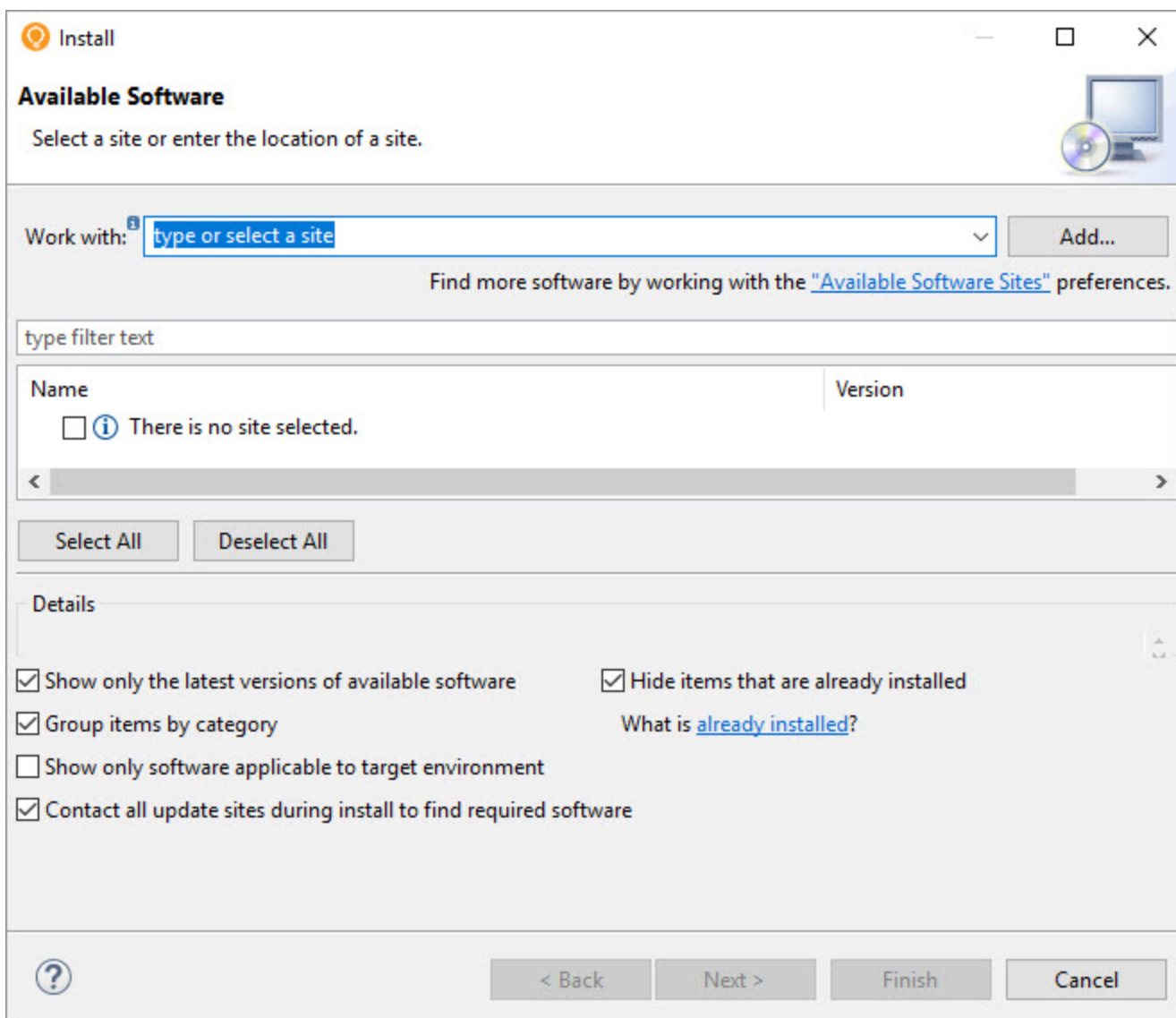
Follow these instructions to install the ASG-JOB/SCAN Plug-in after you download the product files.

### *To install ASG-JOB/SCAN Plug-in:*

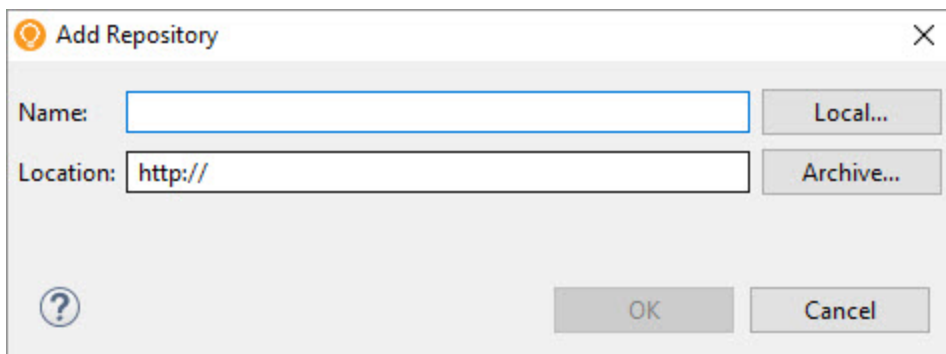
1. Start Topaz Workbench.
2. From the menu, select **Help > Install New Software....**



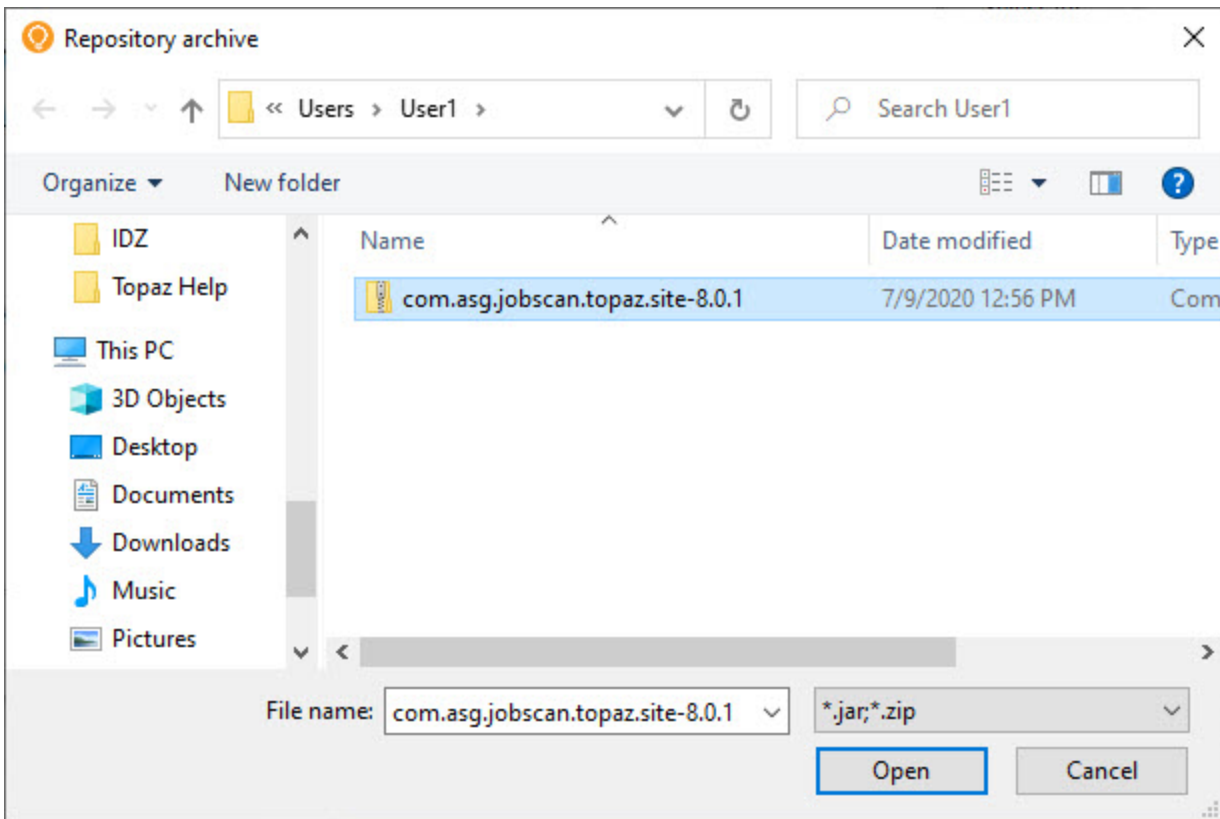
3. In the **Install** dialog, click **Add...**



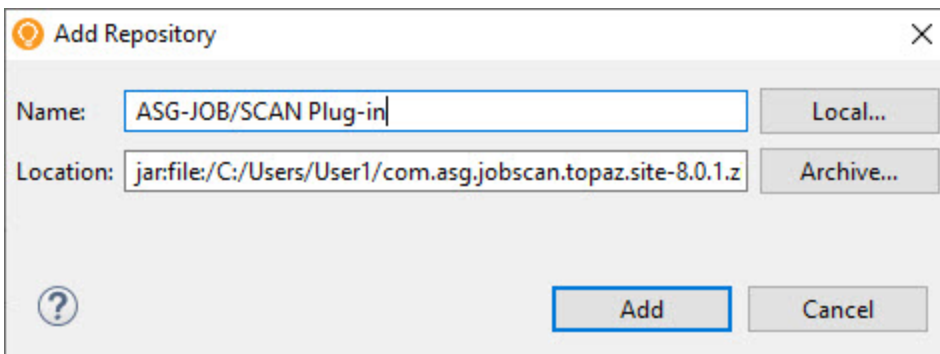
4. In the **Add Repository** dialog, enter a **Name** for the repository and click **Archive...**



5. In the **Repository archive** dialog, browse to select the ZIP file containing the install files, and click **Open**.

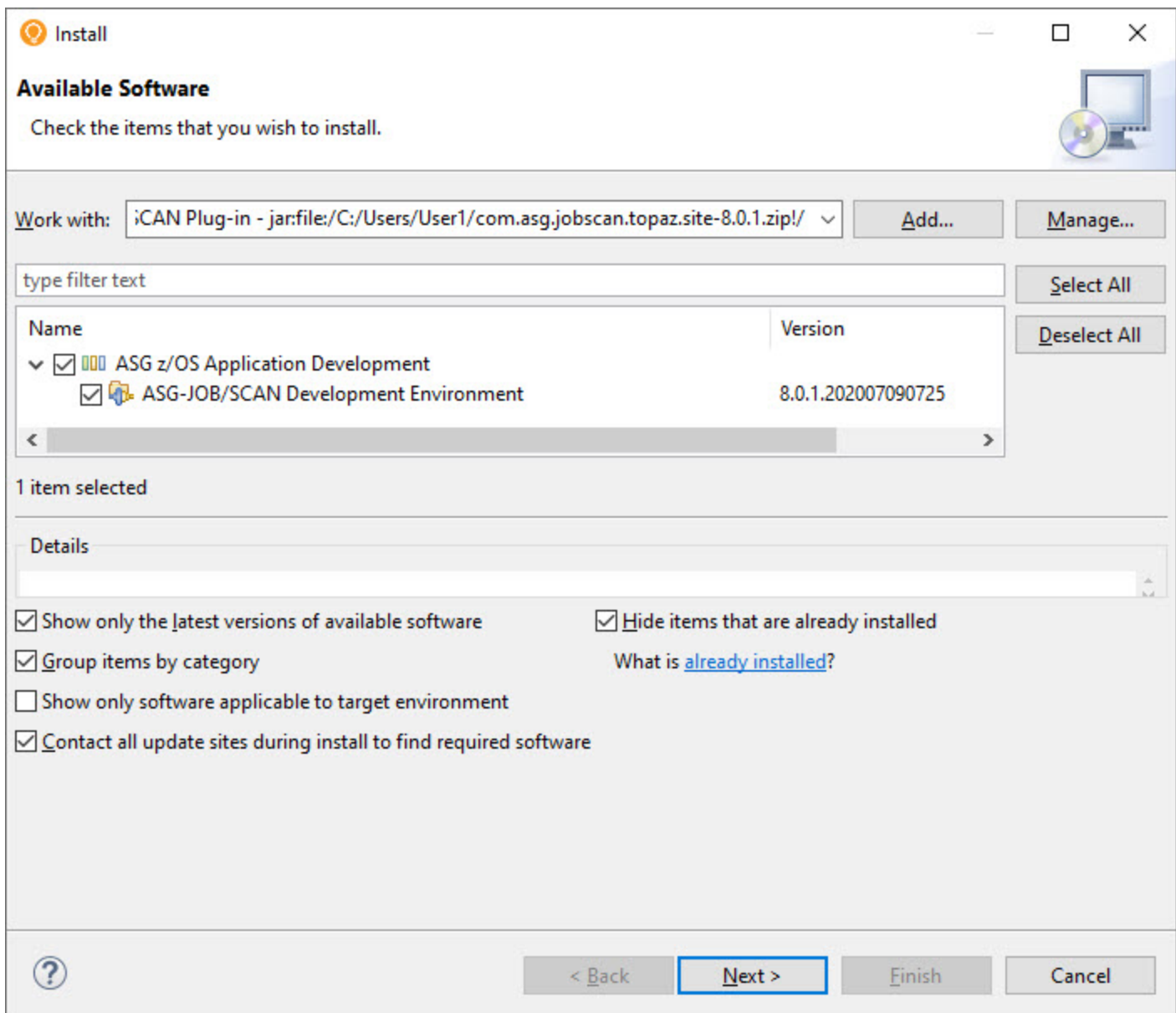


6. In the **Add Repository** dialog, click **Add**.

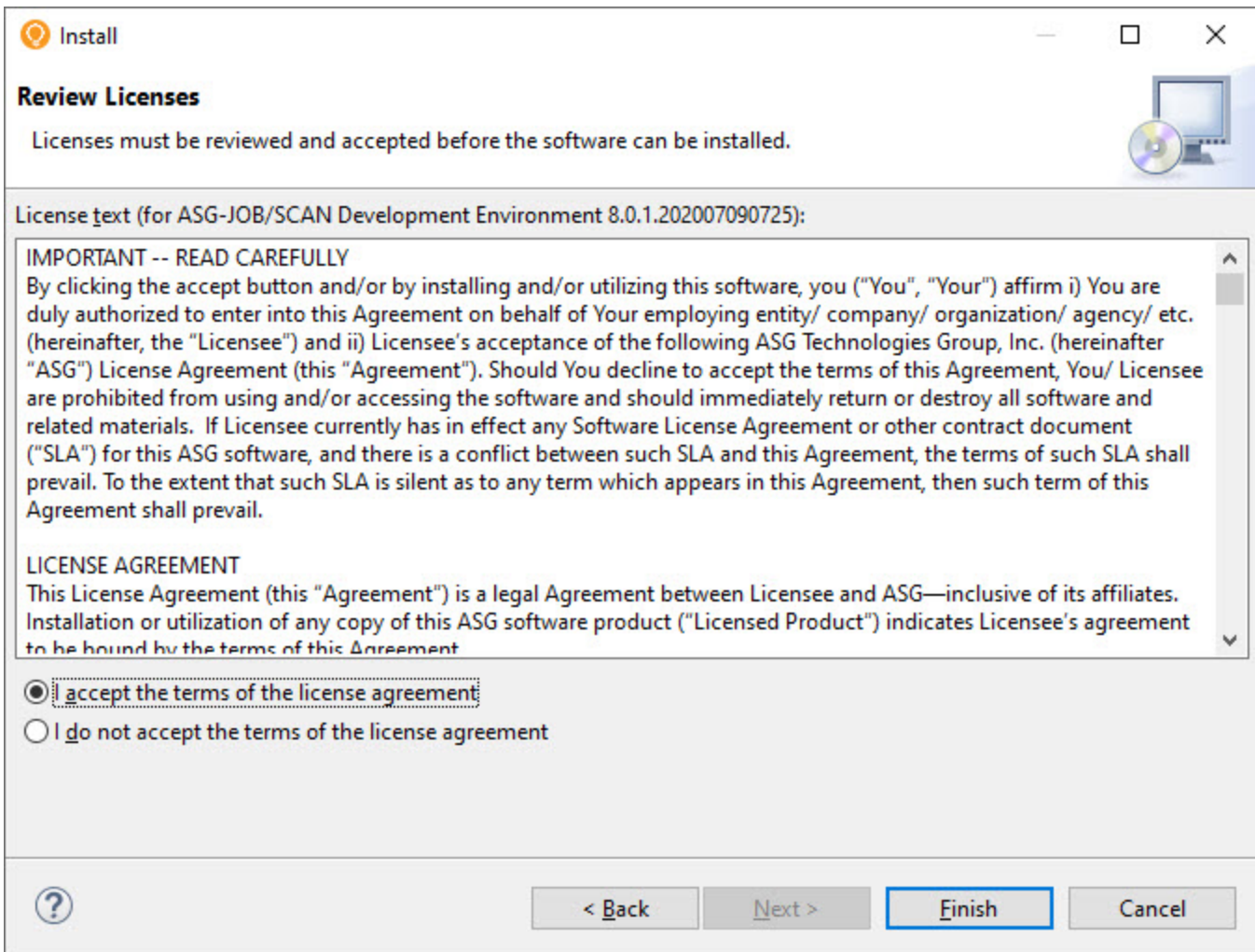


The **ASG z/OS Application Development** category appears in the **Name** area.

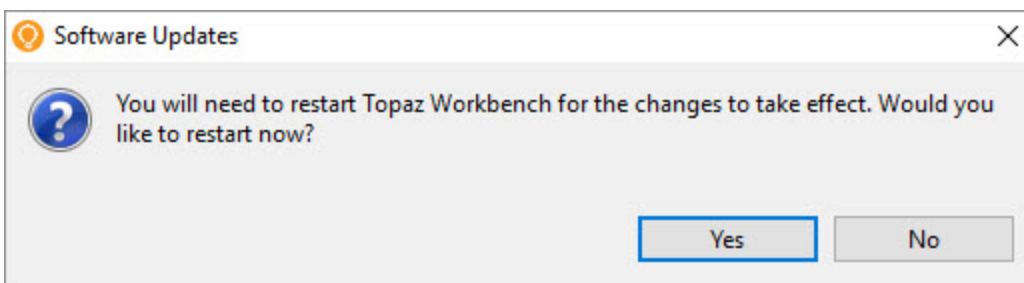
7. Select the **ASG z/OS Application Development** category and make sure that **ASG-JOB/SCAN Development Environment** feature is selected.
8. Click **Next**.



9. Review the install items and click **Next**.
10. Review the license and select **I accept the terms of license agreement**.



11. Click **Finish**. You may receive a security warning. Click **Install anyway** to continue with the installation.
12. When the installation is complete, you are prompted to restart the Topaz application. Click **Restart Now**.

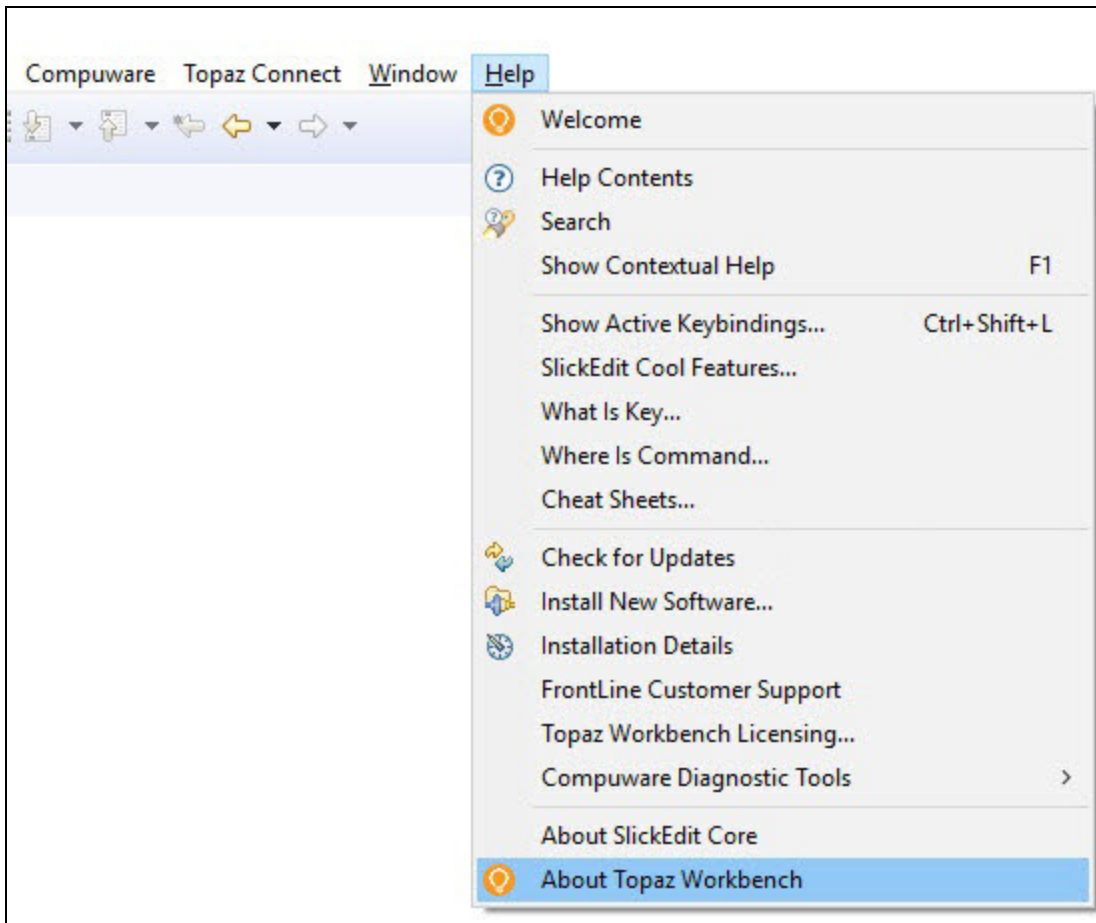


## Verifying the Installation

Follow the instructions in this section to verify whether the ASG-JOB/SCAN Plug-in is installed successfully or to find whether the plug-in is already installed.

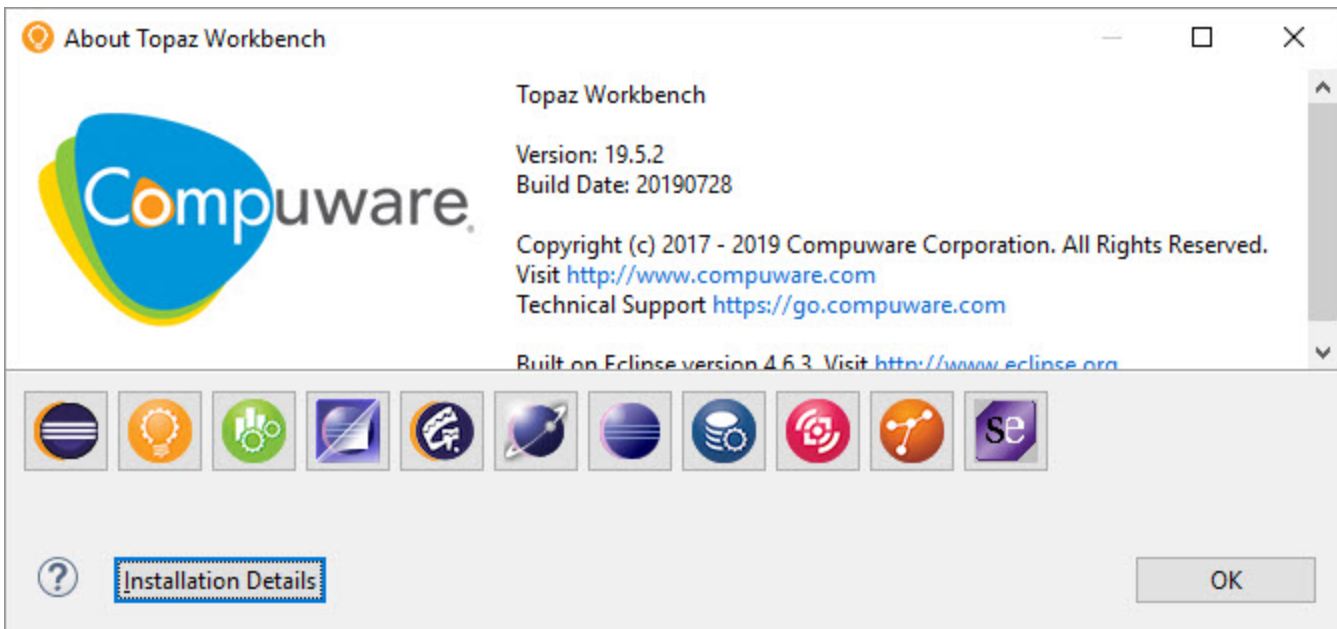
**To verify that the JOB/SCAN Plug-in is installed:**

1. Start Topaz Workbench.
2. From the menu, select **Help > About Topaz Workbench**.

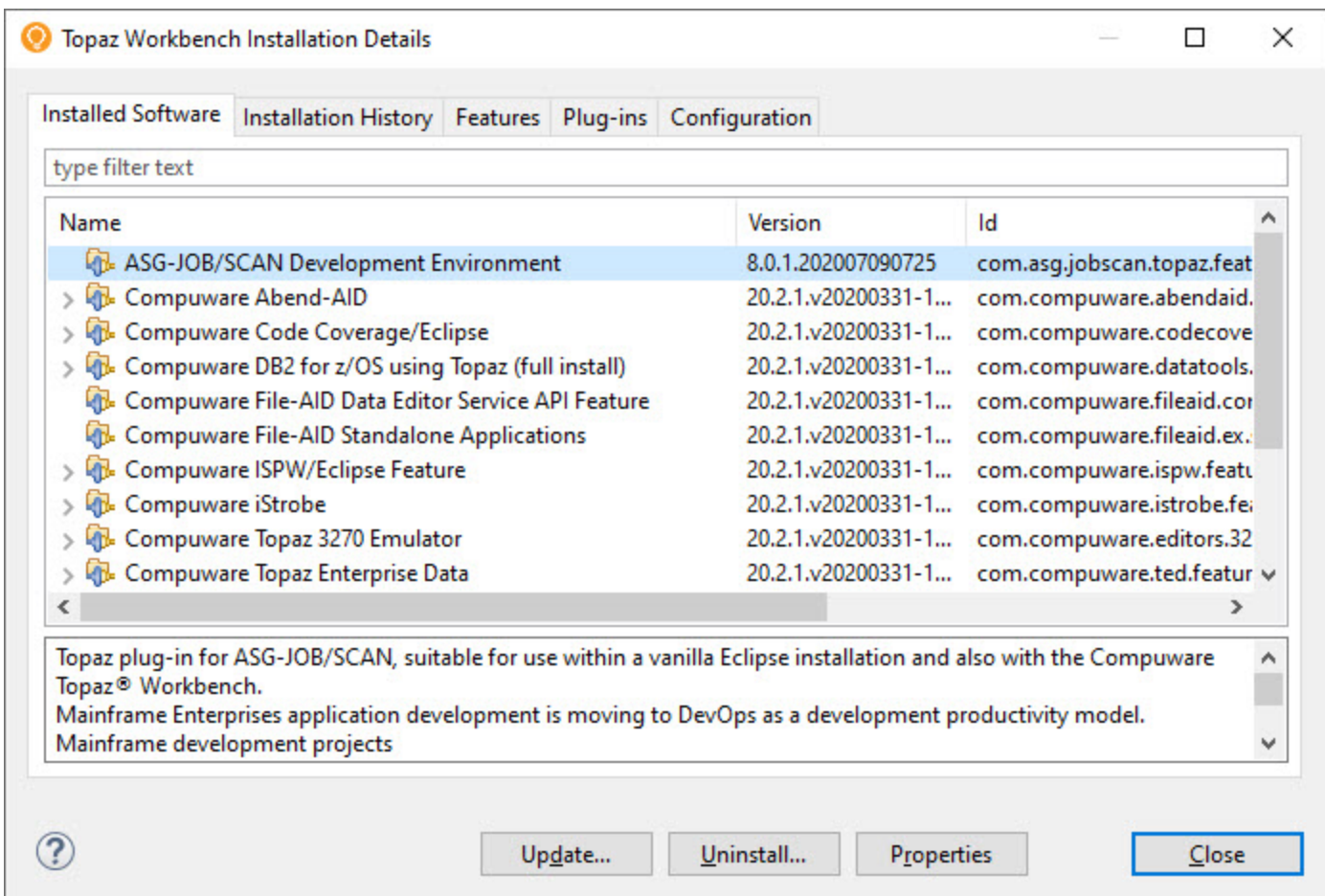


3. In the **About Topaz Workbench** dialog, click **Installation Details**.





- In the **Installed Software** tab, you should see **ASG-JOB/SCAN Development Environment** in the list of installed software. This indicates that JOB/SCAN Plug-in has been installed.



- ★ You can check the **Installation History** tab (Help > About Topaz Workbench > Installation Details > Installation History) for information on when a feature is installed.

## Verifying the z/OSMF Configuration

IBM z/OS® Management Facility (z/OSMF) must be installed and configured before using ASG-JOB/SCAN Plug-in. This topic provides instructions to verify if z/OSMF is configured as per ASG-JOB/SCAN Plug-in requirements.

- ★ This section does not explain the z/OSMF configuration procedure. See [Configuring z/OSMF](#) in IBM Knowledge Center for more information.

### Step to verify z/OSMF Configuration

- Verify if z/OSMF server is running

Issue the console command `/D A, IZU*` in the command line and check if the started tasks jobs, **IZUANG1** and **IZUSVR1**, are running. If the jobs are not active, issue `/S IZUANG1` and `/S IZUSVR1` commands to activate.

- Verify if z/OSMF REST services are enabled

Open a web browser to z/OS system (host name and port) and add the following REST call to the URL: **GET** `/zosmf/restjobs/jobs`. You should see the list of the jobs that are submitted from your user ID.

- Verify if z/OSMF data set and file REST services are enabled

See if the job **IZURFSEC** (available in the sample jobs package) is submitted and completes with the return code '0000'. To verify if services are running, issue a REST service request. See [List the z/OS data sets](#) in IBM Knowledge Center for syntax on how to issue a request.

The job IZUSVR1 shows the actual REST services that are running in z/OSMF. JOB/SCAN Plug-in requires both restjobs and restfiles services. To make sure that these services are available, check the output (STDOUT) files of IZUSRV1 job, and see if it contains a message that looks like:

```
CWWKT0016I: Web application available (default_host): https://<your-zosmf-
url>:<zsmf-port>/zosmf/<service>/
```

where <service> is a REST service name.

For more assistance work with your administrator.

- Verify access to z/OSMF

1. In a browser, open the URL: `https://<hostname>:<port>/zosmf/`

Where:

hostname is the hostname or IP address of the system in which z/OSMF is installed

port is the secure port for the z/OSMF configuration

2. Enter the z/OS USER ID that was used to configure z/OSMF
3. Enter the z/OS PASSWORD associated with the z/OS user ID.
4. Click LOGIN.

The Welcome page of IBM z/OS Management Facility is displayed. This confirms that you are authenticated to access z/OSMF.

## References:

- What setup is needed for z/OSMF - [https://www.ibm.com/support/knowledgecenter/SSLTBW\\_2.3.0/com.ibm.zos.v2r3.izua300/IZUHPINFO\\_WhatSetupIsNeeded.htm](https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.izua300/IZUHPINFO_WhatSetupIsNeeded.htm)
- Software prerequisites for z/OSMF - [https://www.ibm.com/support/knowledgecenter/en/SSLTBW\\_2.2.0/com.ibm.zos.v2r3.izua300/IZUHPINFO\\_SoftwarePrereqs.htm](https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.2.0/com.ibm.zos.v2r3.izua300/IZUHPINFO_SoftwarePrereqs.htm)
- Security configuration requirements for z/OSMF - [https://www.ibm.com/support/knowledgecenter/SSLTBW\\_2.3.0/com.ibm.zos.v2r3.izua300/izuconfig\\_SecurityStructuresForZosmf.htm](https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.izua300/izuconfig_SecurityStructuresForZosmf.htm)
- Configuring z/OSMF - [https://www.ibm.com/support/knowledgecenter/SSLTBW\\_2.3.0/com.ibm.zos.v2r3.izua300/IZUHPINFO\\_PartConfiguring.htm](https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.izua300/IZUHPINFO_PartConfiguring.htm)

## Verifying the Liberty Server Configuration

IBM Liberty Server for z/OS must be installed and a Liberty server for ASG-JOB/SCAN REST Services must be configured before using ASG-JOB/SCAN Plug-in. This topic provides instructions to verify if the Liberty server for ASG-JOB/SCAN REST Services is ready as per the ASG-JOB/SCAN Plug-in requirements.



This section does not explain the Liberty Server for ASG-JOB/SCAN REST Services configuration procedure. See [Installer Reference Guide](#) for more information.

A Liberty server instance may be started as a z/OS started task or started from the z/OS UNIX shell. To operate Liberty servers as z/OS started tasks, BBGZSRV by default would be the started task name.

When a Liberty server seeks to use z/OS authorized services, BBGZANGL by default is required. ASG-JOB/SCAN REST Services does not require a Liberty angel process to be setup.



For more information on the various options for configuring and starting a Liberty runtime environment on the z/OS, see the [IBM WebSphere Application Server for z/OS Liberty](#).

## Step to Verify Liberty Server for ASG-JOB/SCAN REST Services

- Verify if Liberty server is running

Issue the console command `/D A, BBGZSRV*` in the command line and check if one or more started tasks **BBGZSRV\*** are running. If you are not using **BBGZSRV\*** sample proc, substitute this name with the name of the “JCL Start proc name” dedicated to ASG-JOB/SCAN REST Services.

- Start a Liberty server

If started task **BBGZSRV\*** or the name of the “JCL Start proc name” dedicated to ASG-JOB/SCAN REST Services is not running you need to start it. The following example shows the command that you would use to start a Liberty server configured for ASG-JOB/SCAN REST Services named “jjsrest” using the sample bbgzsrv proc:

```
START BBGZSRV.JJSREST,PARMS='jjsrest'
```

The following example shows the command that you would use to stop a ASG-JOB/SCAN REST Server (note: z/OS START command syntax of “S membername[.identifier]”, was used to identify jjsrest as the name identifying the task to be started, so you can issues commands against this task)

```
P JJSREST
```

- Review the messages of the Liberty server. If ASG-JOB/SCAN REST Services were successfully started these messages would be logged to the output STDOUT DD.

The following message verifies that JOB/SCAN REST Services is up and running:

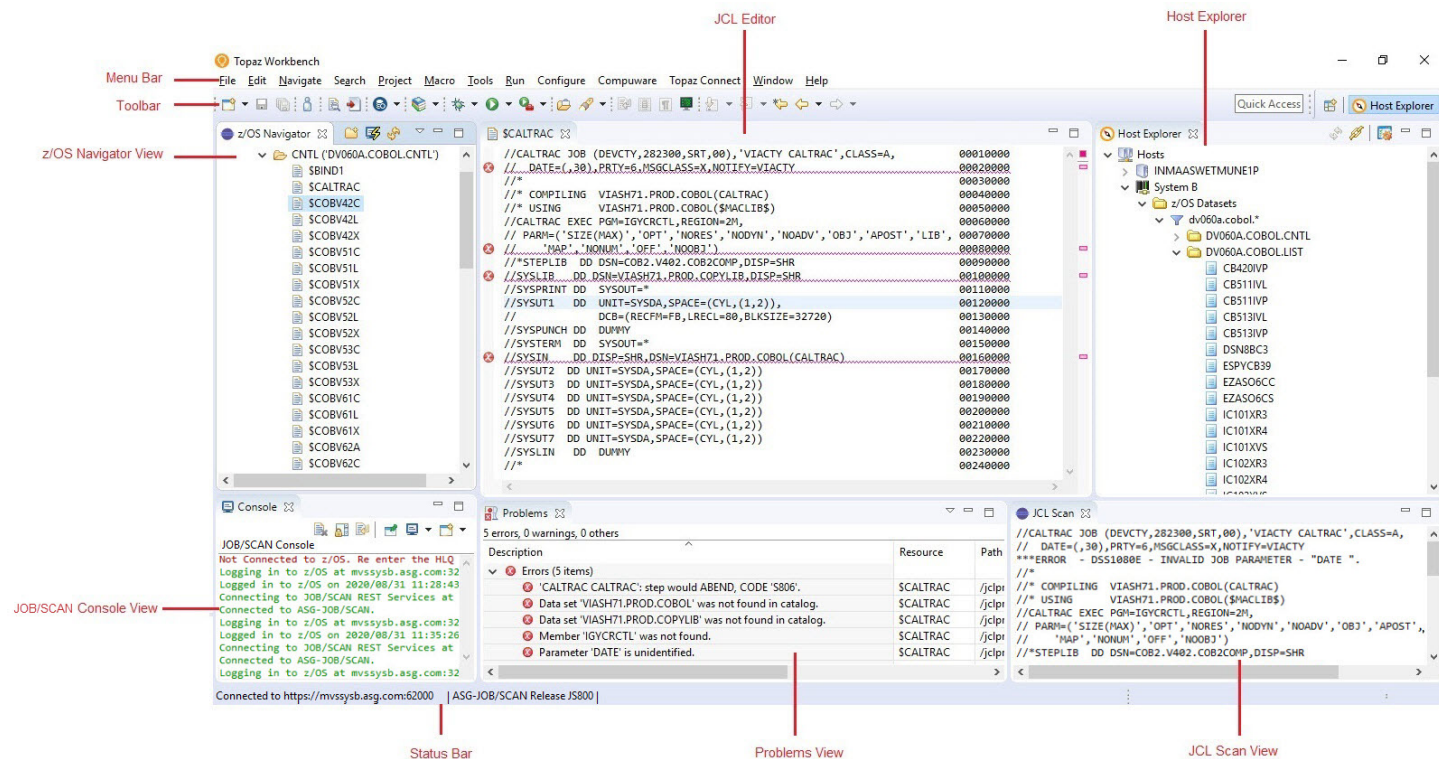
```
CWWKF0011I: The jjsrest server is ready to run a smarter planet.
```

#### Sample STDOUT DD Output

```
Launching jjsrest(WAS FOR Z/OS 19.0.0.12/wlp-1.0.35.cl191220191120-0300) on IBM J9 VM, version
8.0.6.0 - pmz6480sr6-20191107_01(SR6) (en_US)
[AUDIT ] CWWKE0001I: The server jjsrest has been launched.
[AUDIT ] CWWKZ0058I: Monitoring dropins for applications.
[AUDIT ] CWWKT0016I: Web application available (default_host):
http://mvssyse.asg.com:49080/api/docs/
[AUDIT ] CWWKT0016I: Web application available (default_host):
http://mvssyse.asg.com:49080/api/explorer/
[AUDIT ] CWWKT0016I: Web application available (default_host):
http://mvssyse.asg.com:49080/jobscan/
[AUDIT ] CWWKZ0001I: Application jobscan started in 0.556 seconds.
[AUDIT ] CWWKF0015I: The server has the following interim fixes active in the runtime:
PH19989,PH19528. For a full listing of installed fixes run: productInfo version --ifixes
[AUDIT ] CWWKF0012I: The server installed the following features: [distributedMap-1.0, el-3.0,
jaxrs-2.1, jaxrsClient-2.1, jndi-1.0, jsonp-1.1, jsp-2.3, openapi-3.0, servlet-4.0, ssl-1.0,
zosSecurity-1.0].
[AUDIT ] CWWKF0011I: The jjsrest server is ready to run a smarter planet. The jjsrest server
started in 3.358 seconds.
[AUDIT ] CWPKI0820A: The default keystore has been created using the 'keystore_password'
environment variable.
```

# User Interface

This topic describes about the commonly used interface elements that are required when using the JOB/SCAN Plug-in functions. It also provides information on how to view these elements when not displayed in the workbench.



## Menu Bar

The **Menu Bar** contains menus and menu options that are used to perform a task. JOB/SCAN commands are displayed as additional options in the Edit menu. However, you must be connected to JOB/SCAN's server before running the JOB/SCAN commands. Some options in the menu bar are available only when a file is active in the JCL Editor.




## Toolbar

The **Toolbar** provides quick access to tasks that are commonly performed by a user. JOB/SCAN commands are displayed as buttons on the toolbar. However, you must be connected to JOB/SCAN's server before running the JOB/SCAN commands. Some options on the toolbar are available only when a file is active in the JCL Editor.

## z/OS Navigator View

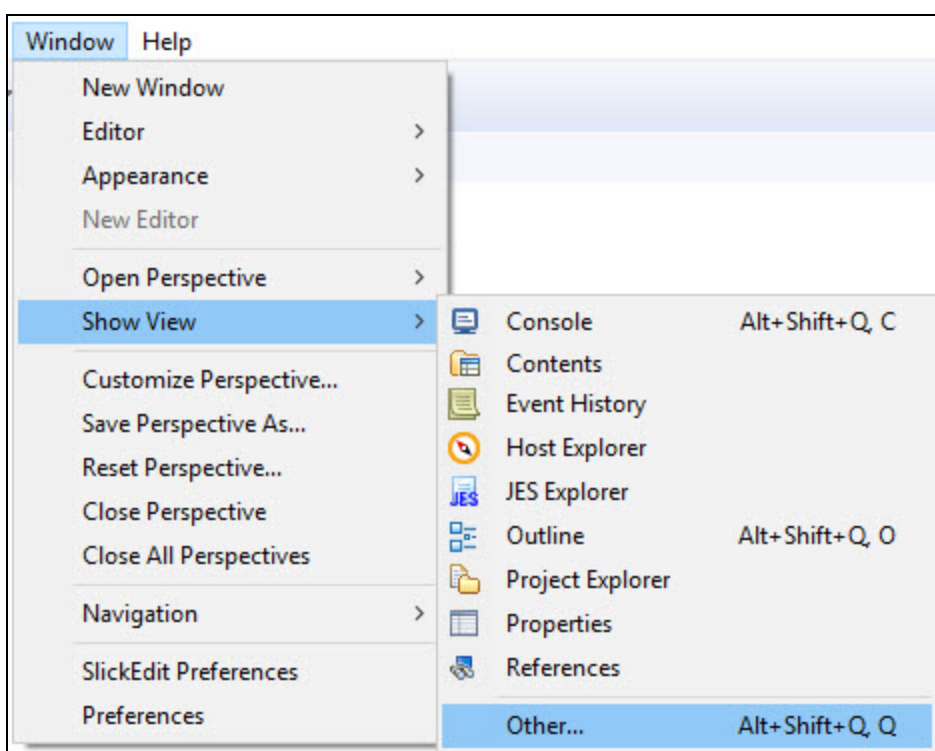
**z/OS Navigator** view displays datasets that are fetched from z/OS. The datasets are displayed under a High-Level Qualifier (HLQ) in a tree-like structure. For more information about displaying datasets, refer to [Displaying Datasets](#).

z/OS Navigator View consists of following icons:

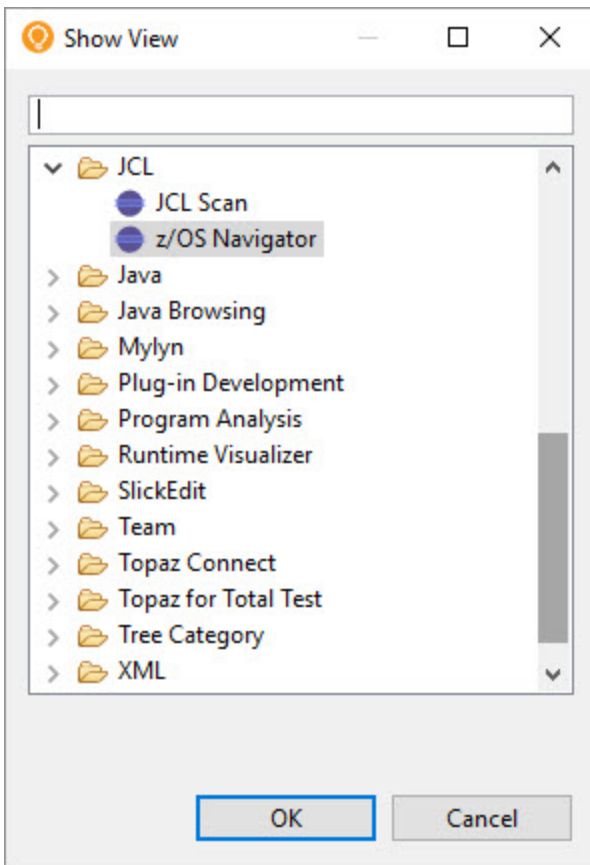
Icon	Name	Description
	Add High Level Qualifier	Fetches datasets from the z/OS system and displays under the specified HLQ. See <a href="#">Working with z/OS Datasets</a> .
	Connect to z/OS	Establishes connection to z/OS system. See <a href="#">Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator</a> .
	Refresh View	Refreshes the list of datasets displayed. See <a href="#">Refreshing Dataset</a> .

### To display z/OS Navigator view:

1. From the menu, select **Window > Show View > Other....**



2. In the **Show View** dialog, do one of the following:
  - Enter **z/OS Navigator** in the **type filter text** box.
  - Locate and expand the **JCL** folder.



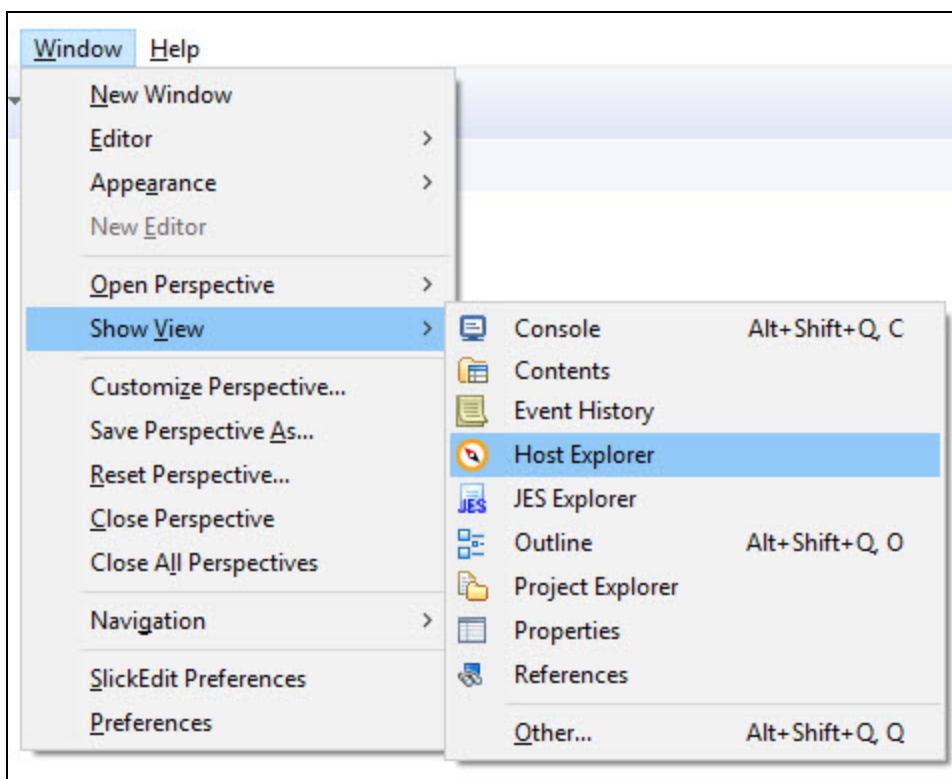
3. Double-click on **z/OS Navigator** or select **z/OS Navigator** and click **Open**.

## Host Explorer

**Host Explorer** view is similar to the z/OS Navigator view. It displays datasets that are fetched from z/OS system and provides hierarchical navigation among them. Host Explorer also provides navigation for local Windows files and various types of distributed relational databases.

### *To display Host Explorer view:*

- From the menu, select **Window > Show View > Host Explorer**.



## Status Bar

The **Status Bar** displays the z/OS connection status.

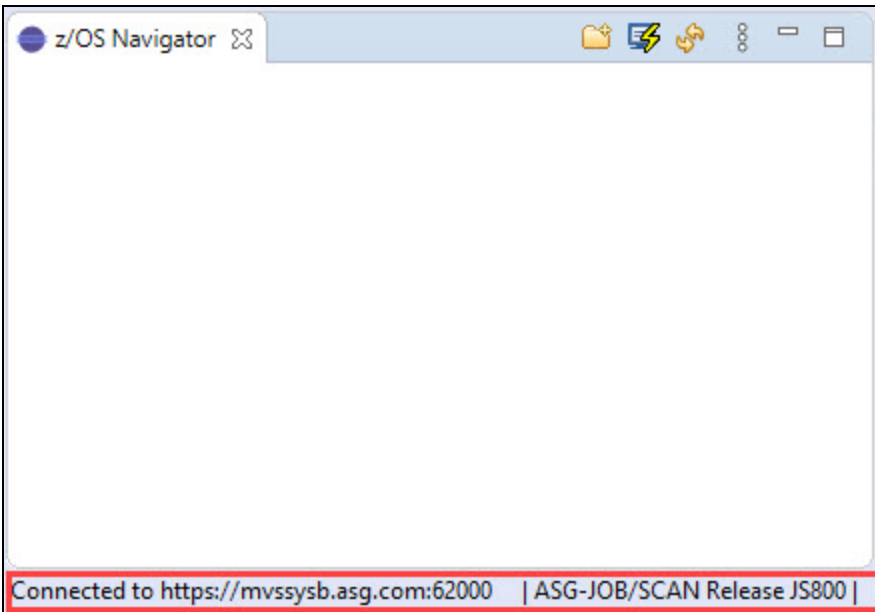


The z/OS connection status is shown only when the **z/OS Navigator** view is active.

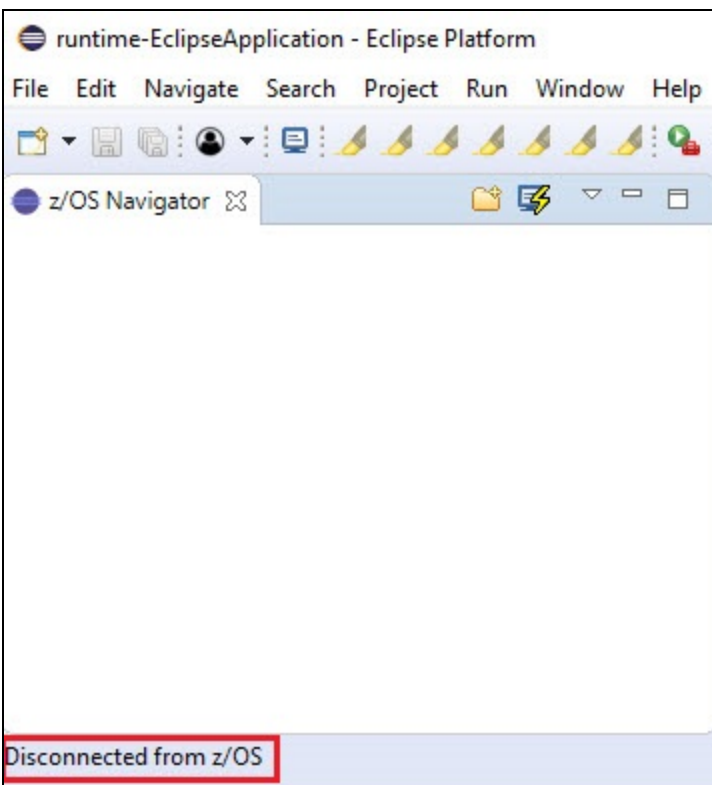
After establishing the connection to z/OS and JOB/SCAN server, the status bar displays the following message:

**Connected to** <host address or IP address of the z/OS system>:<port number of the host machine on which the JOB/SCAN REST API service is running> | **ASG-JOB/SCAN Release JS**<version of the ASG-JOB/SCAN it is connected> |





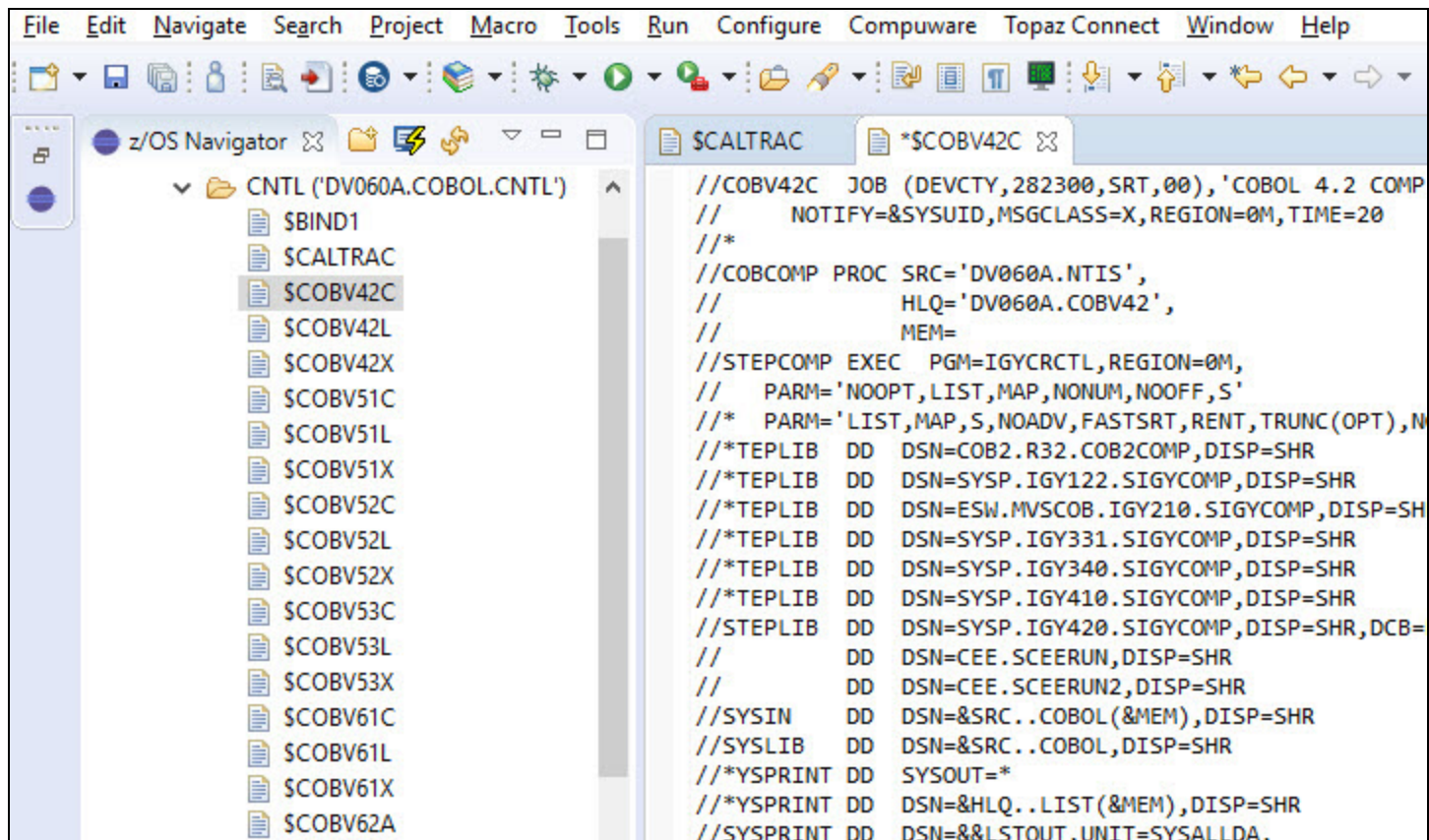
When the connection is lost or when the session is disconnected from z/OS, the status bar display the message: **Disconnected from z/OS**.



## JCL Editor

The **JCL Editor** is typically used to view, edit, save and validate a JCL file. You can view more than one file in the Editor area, but only one can be active at a time. Tabs in the editor area indicate the names of files that are currently open for

editing. Some options in the menu bar and toolbar are available only when a file is active in the JCL Editor.



## Problems View

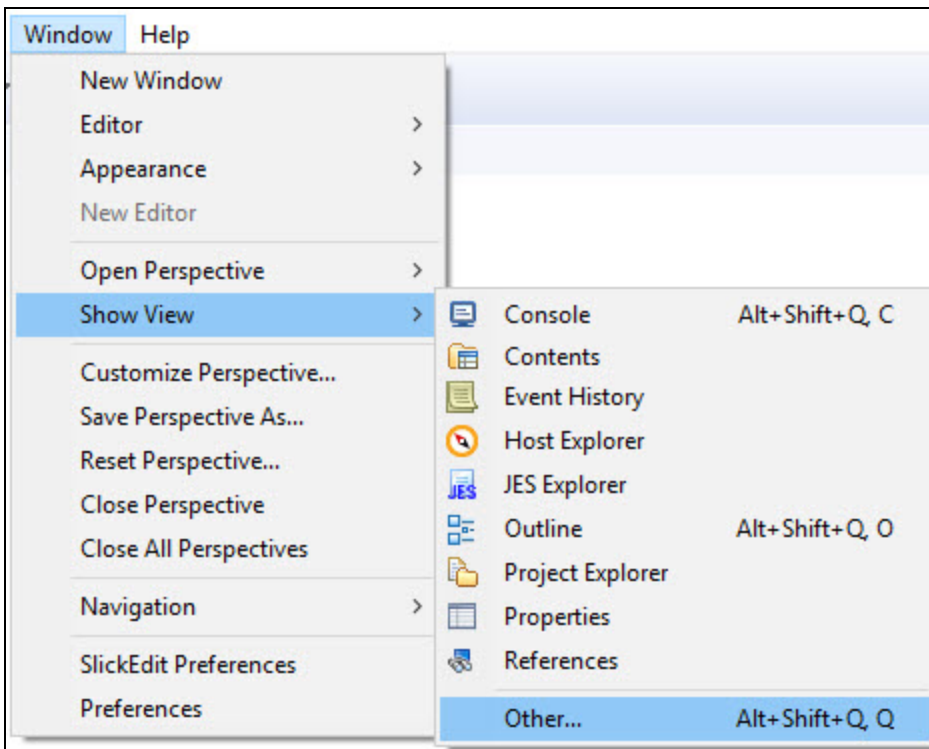
**Problems** view displays the list of problem messages that are identified in a JCL file. Generally, this view is automatically displayed when you scan a file. For more information about Problems view, refer to [Reports](#).

The screenshot shows the Problems view window. The title bar indicates '2 errors, 0 warnings, 401 others (Filter matched 102 of 403 items)'. The table below lists the messages:

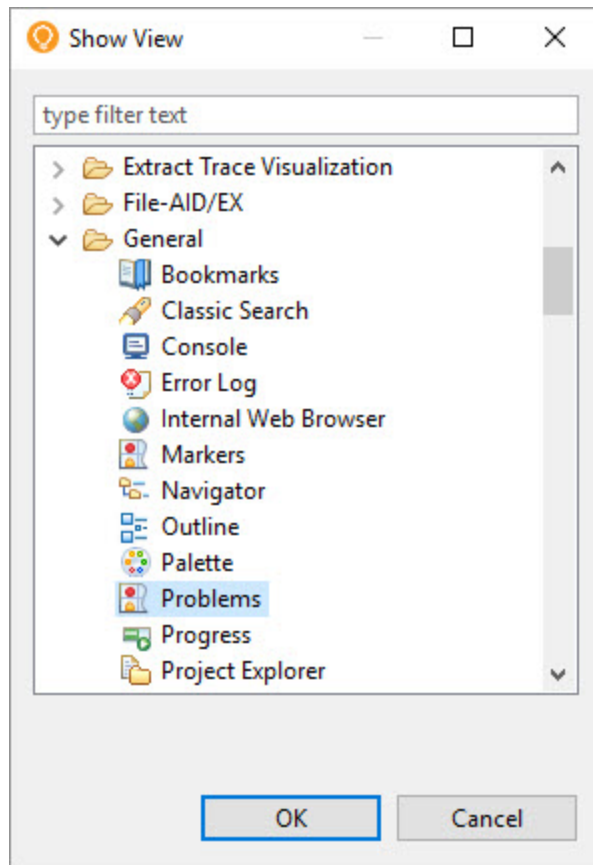
Description	Resource	Path	Location	Type
✖ Errors (2 items)				
✖ DATA SET NOT FOUND IN CATALOG	SCALTRAC	/jclproject/DV060A....	line 10	JCL Scan Mes...
✖ INVALID JOB PARAMETER - "DATE ".	SCALTRAC	/jclproject/DV060A....	line 2	JCL Scan Mes...
ℹ Infos (100 of 401 items)				
ℹ DSN = "VIASH71.PROD.COPYLIB ""},{ "error-id	SCALTRAC	/jclproject/DV060A....	line 10	JCL Scan Mes...
ℹ TEMPORARY DATA SET WILL BE USED.	\$COBV42C	/jclproject/DV060A....	line 65	JCL Scan Mes...
ℹ TEMPORARY DATA SET WILL BE USED.	\$COBV42C	/jclproject/DV060A....	line 65	JCL Scan Mes...
ℹ TEMPORARY DATA SET WILL BE USED.	\$COBV42C	/jclproject/DV060A....	line 65	JCL Scan Mes...
ℹ TEMPORARY DATA SET WILL BE USED.	\$COBV42C	/jclproject/DV060A....	line 65	JCL Scan Mes...

### To display Problems view:

1. From the menu, select **Window > Show View > Other....**



2. In the **Show View** dialog, do one of the following:
  - Enter **Problems** in the **type filter text** box.
  - Locate and expand the **General** folder.



- iii. Double-click on **Problems** or select **Problems** and click **Open**.

## JCL Scan View

**JCL Scan** view displays the scanned report of a JCL file. This report is similar to the JOB/SCAN's Structured JCL Report (SJL). For more information about JCL Scan view, refer to [Reports](#).

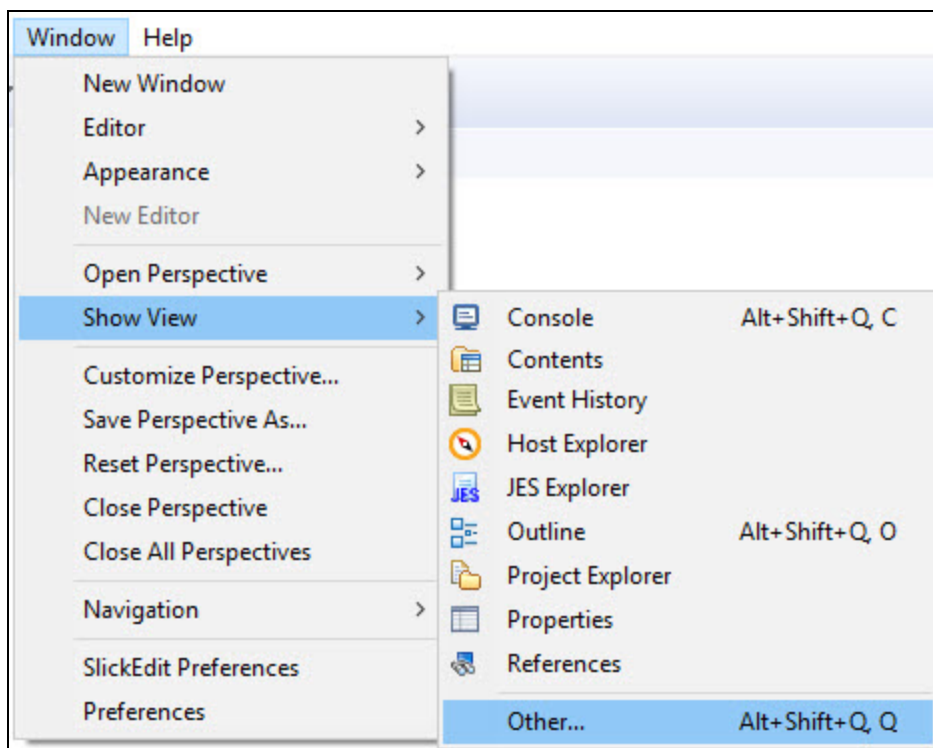
```

JCL Scan
//CALTRAC JOB (DEVCTY,282300,SRT,00), 'VIACTY CALTRAC',CLASS=A,          00010000
// DATE=(,30),PRTY=6,MSGCLASS=X,NOTIFY=VIACTY                        00020000
***ERROR - DSS1080E - INVALID JOB PARAMETER - "DATE ".
//*                                                                    00030000
//* COMPILING VIASH71.PROD.COBOL(CALTRAC)                            00040000
//* USING VIASH71.PROD.COBOL($MACLIB$)                              00050000
//CALTRAC EXEC PGM=IGYCRCTL,REGION=2M,                                00060000
// PARM=('SIZE(MAX)', 'OPT', 'NORES', 'NODYN', 'NOADV', 'OBJ', 'APOST', 'LIB', 00070000
// 'MAP', 'NONUM', 'OFF', 'NOOBJ')
//*STEPLIB DD DSN=COB2.V402.COB2COMP,DISP=SHR                        00090000
//SYSLIB DD DSN=VIASH71.PROD.COPYLIB,DISP=SHR                        00100000
***ERROR - DSS4050E - DATA SET NOT FOUND IN CATALOG
*ADVISORY - DSS8900A - DSN = "VIASH71.PROD.COPYLIB
//SYSPRINT DD SYSOUT=*                                              00110000
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,2)),                            00120000
// DCB=(RECFM=FB.LRECL=80.BLKSIZE=32720)                            00130000

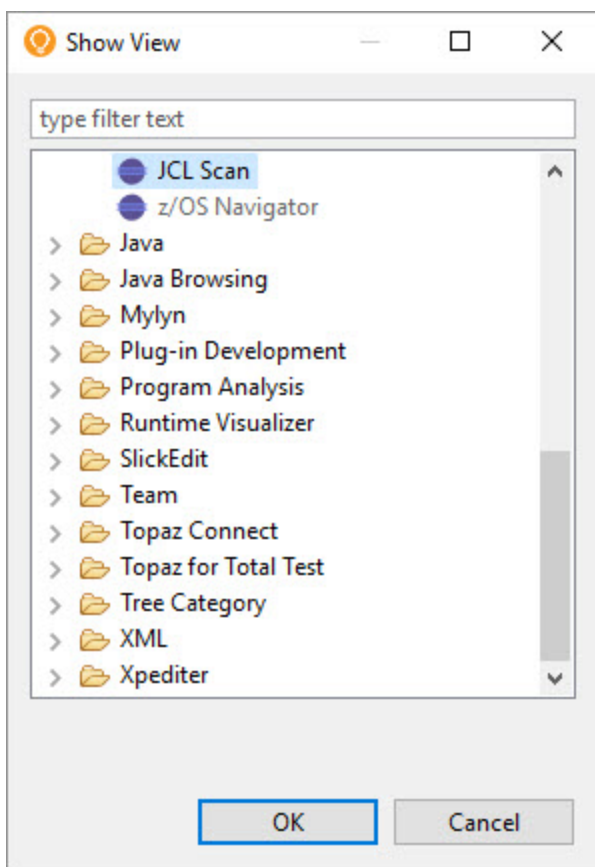
```

### To display JCL Scan view:

1. From the menu, select **Window > Show View > Other...**



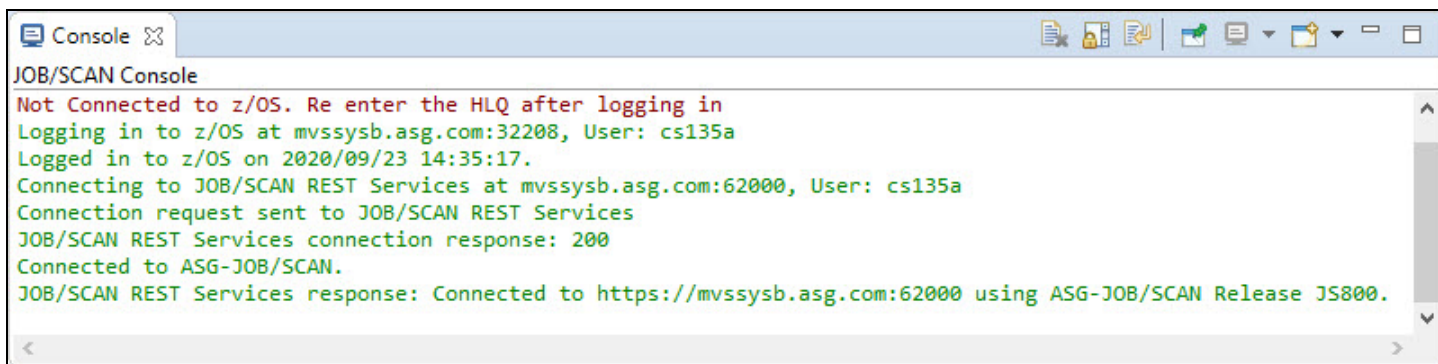
2. In the **Show View** dialog, do one of the following:
  - Enter **JCL Scan** in the **type filter text** box.
  - Locate and expand the **JCL** folder.



3. Double-click on **JCL Scan** or select **JCL Scan** and click **Open**.

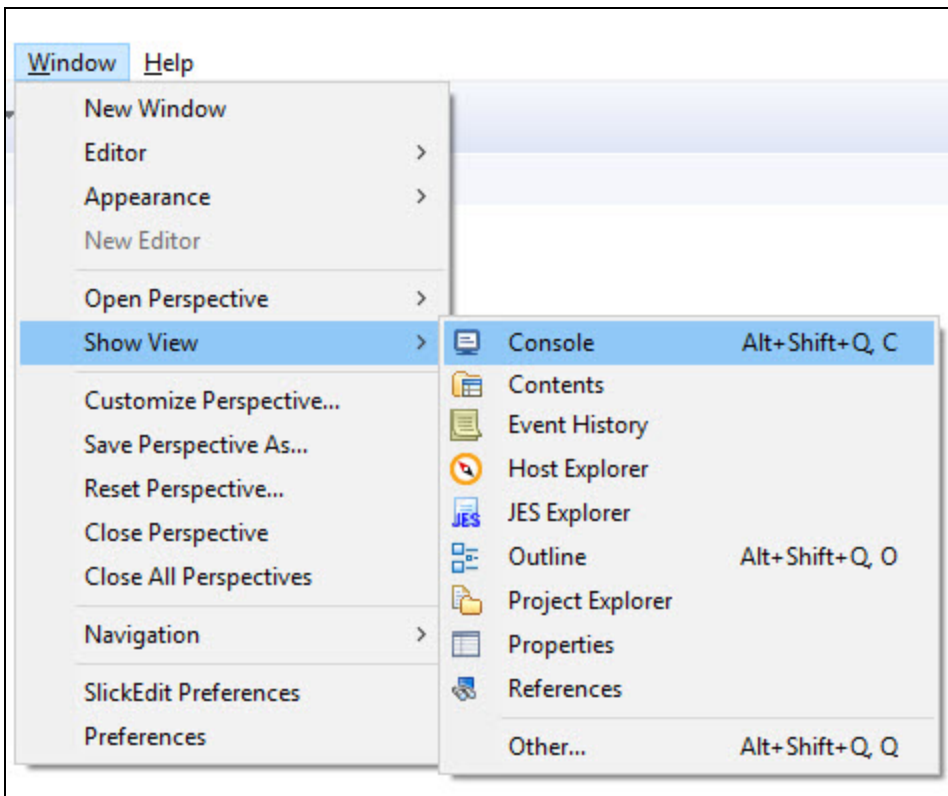
## JOB/SCAN Console View


**JOB/SCAN Console** view displays the output of a process performed in JOB/SCAN Plug-in. The console displays the output messages in two different colors: **Green** for the success messages and **red** for the error messages.



### To display JOB/SCAN Console view:

1. From the menu, select **Window > Show View > Console**.



2. In **Console** dialog, click **Open Console** icon (  ) and select **JOB/SCAN Console**.



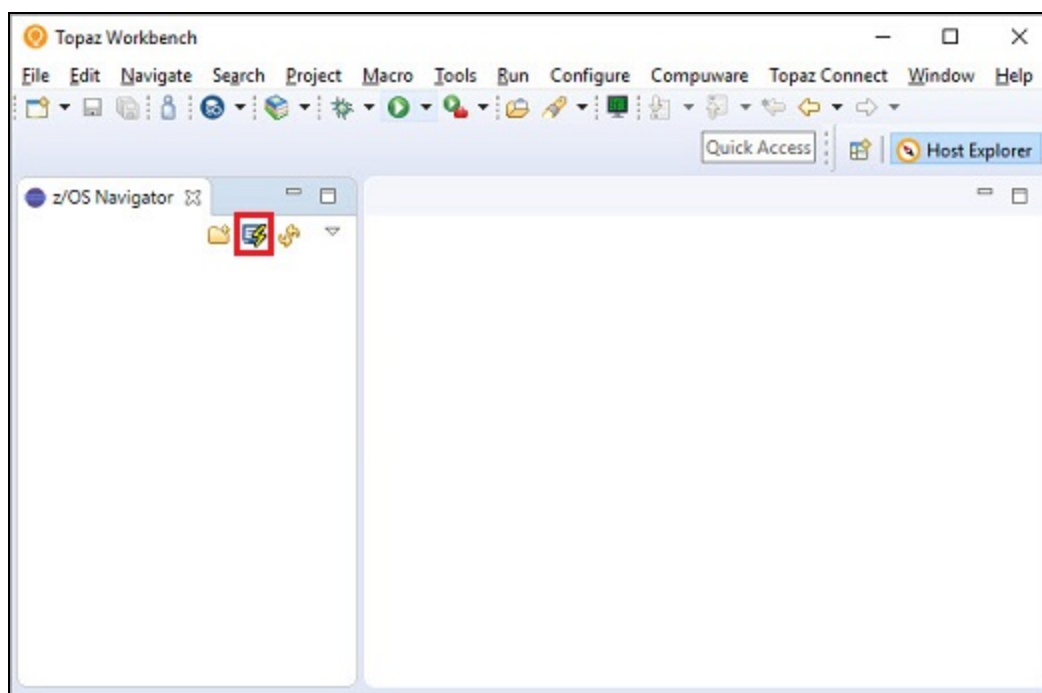
# Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator

This topic explains the connection to z/OS system and ASG-JOB/SCAN. Connection to z/OS and JOB/SCAN is essential to view z/OS resources and perform JOB/SCAN functionalities.

If you are a first time user, then you must create a connection profile that includes information about the z/OS system and ASG-JOB/SCAN.

## **To connect to z/OS:**

1. Select  (**Connect to z/OS**) icon from the **z/OS Navigator** pane.



The **Connect to z/OS** dialog appears.



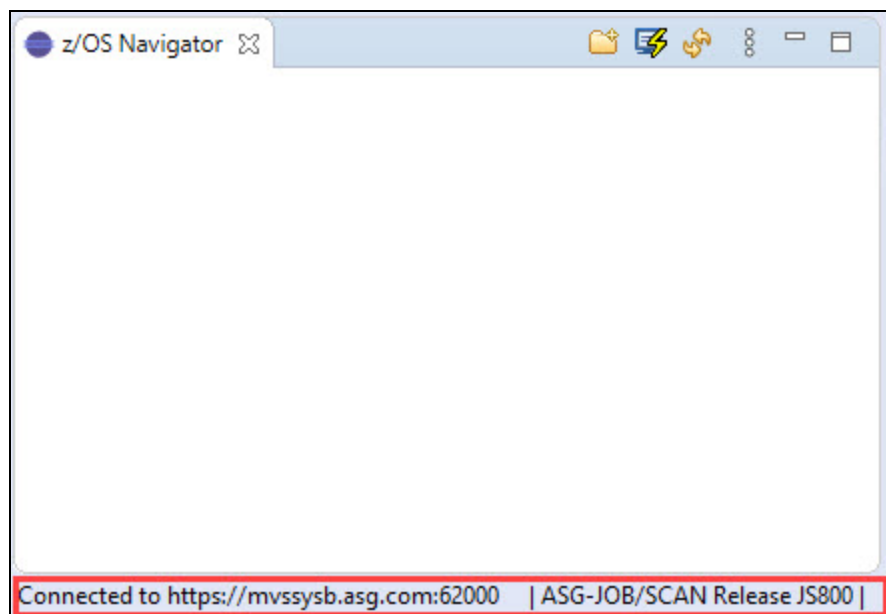
2. Select **System Name** from the list of connection profiles. **MVS User ID** is auto-populated on selecting the system name.




Initially, the **System Name** is empty. Create a new connection profile to display your profile in the **System Name** list. See the sub-topic: [To create a connection Profile](#) for more information.

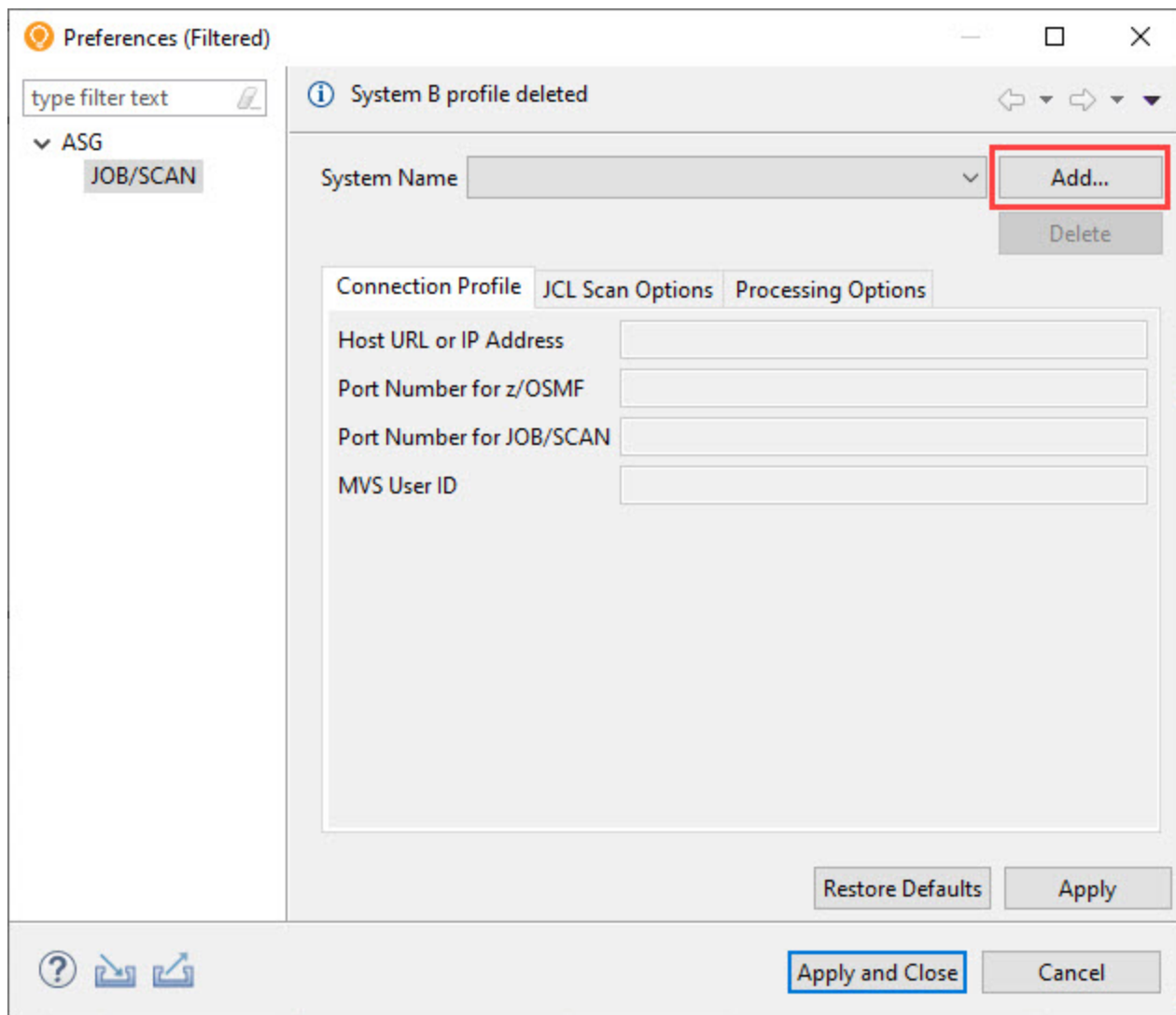
3. Enter **Password** and click **Connect**.

If logon is successful, the **Connect to z/OS** dialog is closed and the connection status along with the version of the JOB/SCAN is displayed in the status bar. Else, a self-explanatory error message appears in the **Connect to z/OS** dialog.

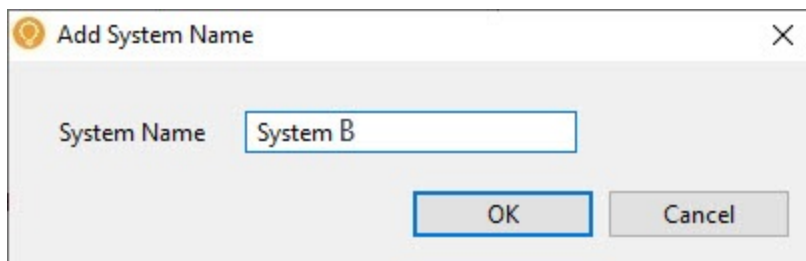


**To create a Connection Profile:**

1. Do either of the following to navigate to the **JOB/SCAN** preference page:
  - From the **z/OS Navigator** pane, select  (Connect to z/OS) icon. In the **Connect to z/OS** dialog, click **Manage Profiles**.
  - From the **Window** menu, select **Preferences**. In the **Preferences** dialog, select **ASG > JOB/SCAN**.
2. Do the following to create a new system name:
  - i. Click **Add...**. The **Add System Name** dialog appears.



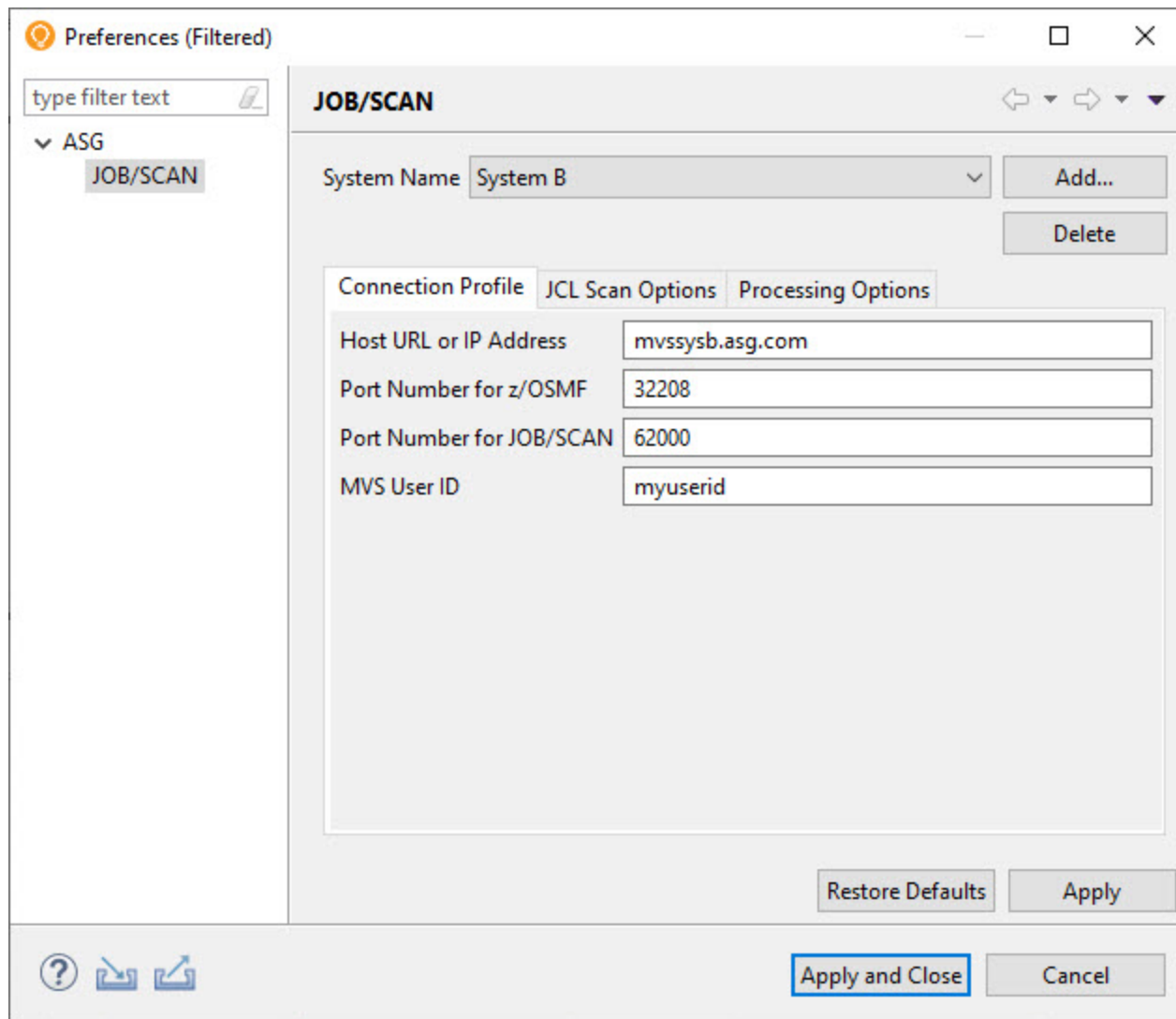
- ii. Enter a name for the new connection profile and click **OK**.



A dialog box titled "Add System Name" with a close button (X) in the top right corner. It contains a text input field labeled "System Name" with the text "System B" entered. Below the input field are two buttons: "OK" and "Cancel".

The given system name appears in the **JOB/SCAN** preference page.

3. In the **Connection Profile** tab fill the following details:



The "Preferences (Filtered)" dialog box shows the "JOB/SCAN" section. On the left, a tree view shows "ASG" expanded with "JOB/SCAN" selected. The main area has a "System Name" dropdown menu set to "System B", with "Add..." and "Delete" buttons to its right. Below this are three tabs: "Connection Profile" (selected), "JCL Scan Options", and "Processing Options". Under the "Connection Profile" tab, there are four text input fields: "Host URL or IP Address" (mvssysb.asg.com), "Port Number for z/OSMF" (32208), "Port Number for JOB/SCAN" (62000), and "MVS User ID" (myuserid). At the bottom right of the dialog are "Restore Defaults" and "Apply" buttons. At the very bottom of the dialog are "Apply and Close" and "Cancel" buttons.

Field Name	Description
Host Name or IP Address	Enter host address or IP address of the z/OS system.

Field Name	Description
Port Number for z/OSMF	Enter port number of the host machine on which the z/OSMF REST API service is running.
Port Number for JOB/SCAN	Enter port number of the host machine on which the JOB/SCAN REST API service is running.
MVS User ID	Enter User ID for logging on to the z/OS system.

4. Click **Apply** to save the scan options or click **Apply and Close** to save the scan options and return to the **Connect to z/OS** dialog.

**To update a Connection Profile:**

1. In the **JOB/SCAN** preference page, select the **System Name** that you like to edit and update the details.
2. Click **Apply** to save the connection profile or click **Apply and Close** to save the connection profile and return to the **Connect to z/OS** dialog.



After updating the connection profile, save the changes and reconnect to ASG-JOB/SCAN using the Connect to z/OS dialog to establish connection with updated details.

**To delete a Connection Profile:**

1. In the **JOB/SCAN** preference page, select the **System Name** that you like to delete and click **Delete**.
2. In the confirmation dialog, click **OK**. Your connection profile is deleted from the **System Name** list.



The connection profile details and scan options that are saved in the System Name will be lost, so make sure before you delete it.

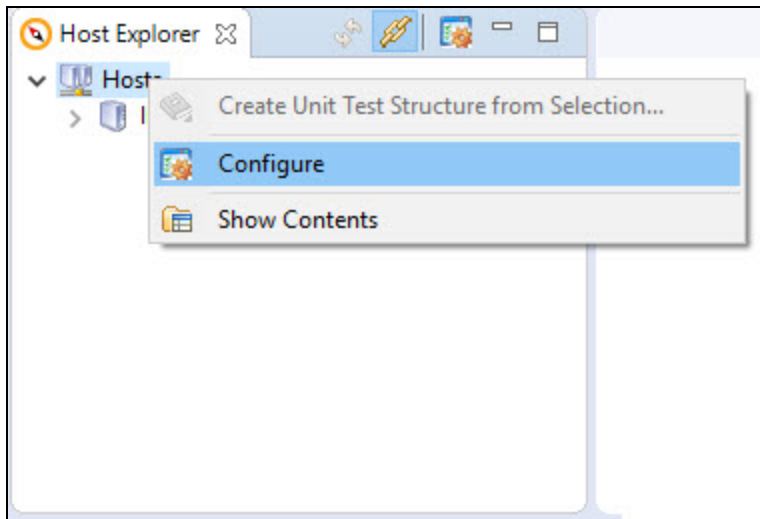
# Connecting to z/OS from Host Explorer

This topic explains the connection to z/OS system from Host Explorer view.

★ Host Explorer view which is a part of the Host Explorer perspective is developed by Compuware, so the sample UI presented in this guide may look different and it is only for illustrative purpose.

## To connect to z/OS:

1. In the **Host Explorer**, right-click on the **Hosts** and select **Configure**.



The **Preferences** dialog box appears with **Host Connections** page.



Host Connections

**New Host Connection**

Create a new host connection

Compuware

General | Credentials | Advanced

Description: System B

Host: mvssysb.asg.com

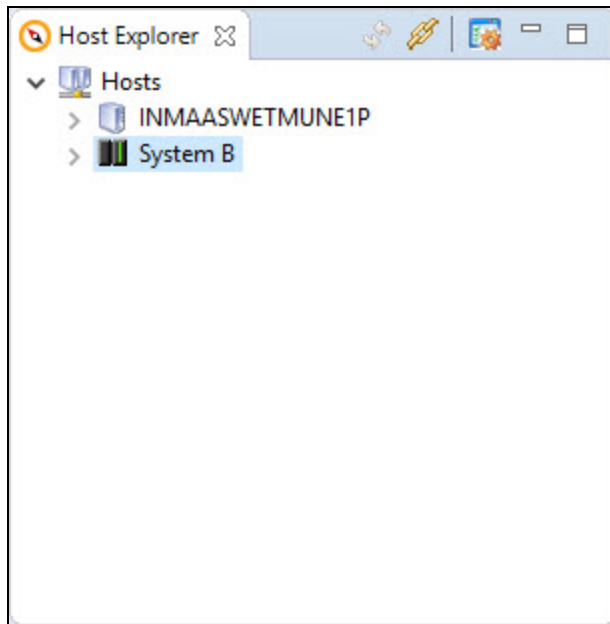
Port: 55197

Encryption protocol: None

OK Cancel

Field Name	Description
Description	Enter a name for the new HCI connection.
Host	Enter URL or IP address of the z/OS system that is running HCI.
Port	Enter Port number to which HCI is assigned.
Encryption protocol	From the list, select the type of encrypted secure HCI connection.

The new host connection appears in the **Host Explorer** view.



4. Expand the new host connection. You will be prompted for mainframe logon credentials.

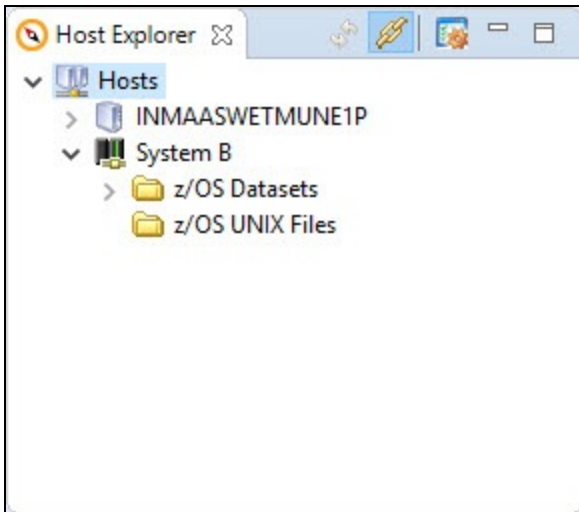
A screenshot of a "Login" dialog box titled "Login Credentials for System B". The dialog prompts the user to enter login credentials. It contains the following fields and options:

- User ID: [Text input field]
- Password: [Text input field]
- Save credentials
- Additional Parameters** (expanded):
  - Group ID: [Text input field]
  - Host code page: 1047 - Latin 1 (Open Systems) [Dropdown menu]
- Change Password** (collapsed)

At the bottom right, there are "OK" and "Cancel" buttons.

5. Enter TSO **User ID** and **Password**.
6. Click **OK**. The **Host Explorer** displays two folders under the new host connection, namely, **z/OS Datasets** and **z/OS UNIX Files**. This confirms that your connection is successful.





# Setting the JCL Scan Options

This topic explains how to set scan options that are used when validating JCLs. Similar to ASG-JOB/SCAN, the ASG-JOB/SCAN Plug-in allows you to specify some options that are used when validating the JCLs. You can set the following options in ASG-JOB/SCAN Plug-in:

## First PROCLIB

When validating JCL, it might be necessary to specify a PROC library to be searched before any other libraries. The Scan Options tab provides a First PROCLIB field for this purpose. When this field is set, JOB/SCAN will look for the PROC and INCLUDE members used in the input JCL file in the library first before searching other libraries. The only type of library supported in this release is PDS.

## Selection Exit

A Selection Exit is a user-defined program that processes the input JCL before JOB/SCAN parses and validates it. The exit evaluates the JCL, makes changes if necessary, and issues a return code instructing JOB/SCAN on the action to be taken with the JCL. The exits can be written in any LE-compliant language. For sample selection exit programs, refer to CA7J1 and CA7J2 members in the SJJSSAMP library. For more information on Selection Exit, see [Selection Exit Interface](#).

## Standards

In addition to the general error validation, the JOB/SCAN's Standards Facility provides the ability to integrate site's additional requirements for validating JCLs. Standards (standards exits or standards routines) are a set of rules that enforces site-specific requirements into JOB/SCAN's JCL validation. The standards are user-defined programs that can be written in any language including OS/VS COBOL, VS COBOL II, REXX, and Assembler. Sample programs are available in the SJJSTDL library. For a detailed introduction to JOB/SCAN's standards facility, see [Standards Information](#).

### *To set the JCL scan options:*

1. Do either of the following to navigate to the **JOB/SCAN** preference page:
  - From the **z/OS Navigator** pane, select (Connect to z/OS) icon. In the **Connect to z/OS** dialog, click **Manage Profile**.
  - From the **Window** menu, select **Preferences**. In the **Preferences** dialog, select **ASG > JOB/SCAN**.
2. Make sure that the correct **System Name** is selected or add a new one by clicking **Add....**

Preferences (Filtered)

type filter text

ASG

JOB/SCAN

System Name System B Add... Delete

Connection Profile **JCL Scan Options** Processing Options

Host URL or IP Address myssysb.asg.com

Port Number for z/OSMF 32208

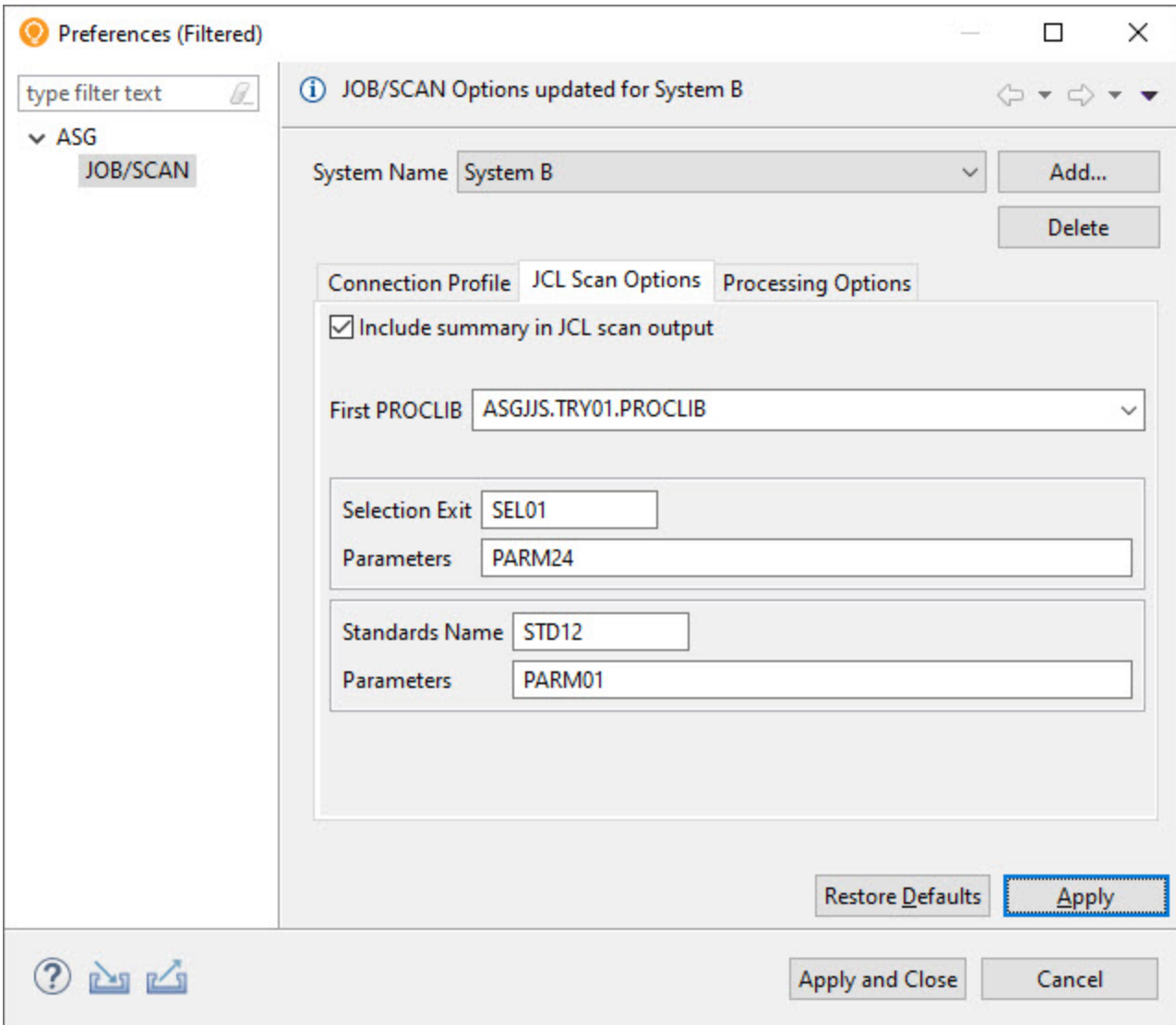
Port Number for JOB/SCAN 62000

MVS User ID myuserid

Restore Defaults Apply

Apply and Close Cancel

3. Make sure that the **Connection Profile** fields are filled in. See [Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator](#) for more information.
4. Select the **JCL Scan Options** tab and fill the following details as per your requirement:



Field Name	Description
Include summary in JCL scan output	Select to display the JCL scan options in the JCL scan report.
First PROCLIB	Enter the library that are searched first before other libraries for PROC and INCLUDE members.
<b>Selection Exit</b>	
Selection Exit	Enter the name of a selection exit program.
Parameter	Enter any parameters that is required for the execution of the selection exit program.

Field Name	Description
<b>Standards</b>	
Standards Name	Enter the name of a standards program.
Parameter	Enter any parameters that is required for the execution of the standards program.

- 
- Click **Apply** to save the scan options or click **Apply and Close** to save the scan options and return to the **Connect to z/OS** dialog.

## Setting the Processing Options

This topic explains the procedure to set the processing options that control how JOB/SCAN processes the input JCL, customize various reports and third-party interfaces. These options are defined in the Preferences page. In ASG-JOB/SCAN, user can enter the values for these options on the Processing Options panel or can code in the SJJSPARM library. Values entered on the panel override corresponding parameters coded in the SJJSPARM library. Similarly, values entered in the Plug-in's Preference page will override corresponding parameters coded in JOB/SCAN's SJJSPARM library. You can set the following processing options:

### Use Error Table

JOB/SCAN provides an option to suppress or change the severity level of JCL error messages for a specified JOB/SCAN error numbers. The entries in the Control Table AW are used to suppress any error messages or change their severity level. When the Use Error Table option is selected, JOB/SCAN searches the Control Table AW to determine if any error messages issued on the JCL statement are listed. If the error number is found in Control Table AW, then the severity is changed to the level indicated or is optionally suppressed.

### Condition Code Processing

JOB/SCAN provides an option to control the effect of CONDCODE parameters during execution. The condition code validation is done for COND= parameters on the job statement, EXEC PROC statement, EXEC PGM statements, and for // IF statements. Control Table AU contains Job steps to be ignored by JOB/SCAN. Control Table AR contains the return codes. When the Condition Code Processing option is selected, then JOB/SCAN excludes all program steps identified in Control Table AU entries, and each step is assigned a return code 0 unless a Control Table AR entry is specified.

### Catalog Access

The Catalog Access option controls the access to the system catalog. When this is selected, JOB/SCAN is allowed to access the system catalog.

### VTOC Access

The VTOC Access option controls the access to the DASD VTOCs. When this is selected, JOB/SCAN is allowed to access the DASD VTOCs.

### PDS Member Verification

The PDS Member Verification option determines whether or not JOB/SCAN can validate partitioned dataset member names. Verification is done when a DSN=dsname(membername) parameter is coded in the JCL DD statement.

### Security Exit DSNAME Check

The Security Exit DSNAME Check option allows JOB/SCAN to detect access authority violations. When this option is selected, JOB/SCAN instructs the security exit program to verify that the user ID running JCL validation has the necessary access authority to the datasets that are mentioned in the JCL before actually attempting to access the file.



When this option is selected and if there is no security exit active or no DSNNAME-checking support in the exit, then JOB/SCAN ignores this option.

## Program Checking

The Program Checking option determines whether or not JOB/SCAN can validate the existence of programs. If this is selected, JOB/SCAN is allowed to validate the programs.

## Disable SMS Processing

The Disable SMS Processing option determines whether or not JOB/SCAN can access the Automatic Class Selection (ACS) routines and the Construct Access Services (CAS) of the MVS DFSMS subsystem. If this option is selected, JOB/SCAN will bypass all SMS processing. If deselected, the SMS Run option specified in the SJJSPARM library will be in effect.

## IDCAMS Processing

JOB/SCAN has the ability to validate IDCAMS control statements when found within a JCL member. If this option is selected, JOB/SCAN validates IDCAMS control statements and tracks datasets created or deleted by the IDCAMS.

## Use SCHEDULER Interface

When jobs are executed using a scheduler, it allows the JCL to contain environmental variables which must be resolved at the time of job submission. JOB/SCAN supports a wide range of schedulers and it also resolves the environmental variables before the job is submitted. Select this option to turn on the scheduler interface specified for the SCHEDULER Run option in the SJJSPARM library.

## Access Control Cards

A Control Card is a dataset or an inline statement(s) that control how the validation should be processed. When this option is selected, JOB/SCAN is directed to read members within the job stream that contain control cards, verify control card input, and include control cards in the JCL Scan report. You can also limit the number of control cards listed in the JCL Scan view for each DDNAME or dataset name identified as control card input.

## Subsystem

The input JCL may include statements related to Job Entry Subsystem(JES) that need to be processed by JOB/SCAN. This option indicates whether or not JOB/SCAN can validate the JES statements. If JES2 or JES3 is selected, then JOB/SCAN searches for any non-IBM-standard JES statements and uses the entries in Control Table AS to standardize the JES statements.

## Suppress Error Messages

JOB/SCAN provides an option to suppress error messages that are normally displayed in the JCL scan reports. This is done based on the message type. The JOB/SCAN plug-in provides the following options to suppress the messages from being displayed in the JCL Scan view, the Problems view, and the JCL editor:

- **do not suppress:** display all messages
- **advisory only:** suppress advisory messages
- **advisory and warning:** suppress warning and advisory messages
- **advisory, warning, and error:** suppress advisory, warning, and error messages
- **less than or equal to:** suppress all error messages with a severity level equal to or less than the specified value

## Extended Error Text

In JOB/SCAN, you can display a messages in two ways: short message text and extended message text. The short error text is a single-line messages that provide less information about the error. The extended messages text give users a better understanding of the violation beyond what the basic short error text provide. This option allows you to select the type of message text to be displayed for each error in the result of a scan.



For more information on processing options, refer to [ASG-JOB/SCAN Reference Information](#).

### *To set the JCL scan options:*

1. Do either of the following to navigate to the **JOB/SCAN** preference page:
  - From the **z/OS Navigator** pane, select (Connect to z/OS) icon. In the **Connect to z/OS** dialog, click **Manage Profiles**.
  - From the **Window** menu, select **Preferences**. In the **Preferences** dialog, select **ASG > JOB/SCAN**.
2. Make sure that the correct **System Name** is selected or add a new one by clicking **Add...**



Preferences (Filtered)

type filter text

ASG

JOB/SCAN

JOB/SCAN Options updated for System B

System Name System B Add... Delete

Connection Profile JCL Scan Options **Processing Options**

Host URL or IP Address myssysb.asg.com

Port Number for z/OSMF 32208

Port Number for JOB/SCAN 62000

MVS User ID myuserid

Restore Defaults Apply

Apply and Close Cancel

3. Make sure that the **Connection Profile** fields are filled in. See [Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator](#) for more information.
4. Select the **Processing Options** tab and fill the following details as per your requirement:

Preferences (Filtered)

type filter text

ASG

JOB/SCAN

JOB/SCAN Options updated for System B

System Name: System B

Buttons: Add..., Delete

Connection Profile | JCL Scan Options | Processing Options

Use Error Table  
 Catalog Access  
 PDS Member Verification  
 Program Checking  
 IDCAMS Processing  
 Access Control Cards

Condition Code Processing  
 VTOC Access  
 Security Exit DSNAME Check  
 Disable SMS Processing  
 Use SCHEDULER interface

Control Card Limit: [ ] 1 to 999 or blank for no limit

Job Limit: [ ] 1 to 999 or blank for no limit




Subsystem: none

Suppress Error Messages: do not suppress [ 0 ]

Extended Error Text: short message text

Buttons: Restore Defaults, Apply, Apply and Close, Cancel

Field Name	Description
Use Error Table	Select to access Control Table AW for error suppression based on severity levels.
Condition Cod Processing	Select to request that JOB/SCAN evaluate all the CONDCODE parameters.
Catalog Access	Deselect to inhibit access to the system catalog.
VTOC Access	Deselect to inhibit access to the DASD VTOCs.
PDS Member Verification	Select to verify the existence of PDS members referenced in the JCL.
	<div style="border: 1px solid purple; padding: 5px;"> <p>★ This option is disabled if <b>VTOC Access</b> is not selected.</p> </div>

Field Name	Description
Security Exit DSNNAME Check	Select to invoke the security exit to check each dataset name in the input JCL for possible security violations.
Program Checking	Select to validate the existence of programs.
Disable SMS Processing	Select to bypass all SMS processing.  This option has no effect if the SMS subsystem is not active.
IDCAMS Processing	Select to validate IDCAMS control statements.
Use SCHEDULER Interface	Select to turn on the SCHEDULER Run Option in JOB/SCAN's SJJSPARM library.
Access Control Cards	Select to access control card libraries, verify control card input, and include control cards in the validation.  This option is enabled only if both the <b>Catalog Access</b> and <b>VTOC Access</b> options are selected.
Control Card Limit	Limit the number of control cards (1-999) listed in the JCL Scan view for each DDNAME or dataset name identified as control card input.  This field is enabled only if <b>Access Control Cards</b> is enabled and selected.
Job Limit	Limit the number of jobs to be processed (1-999) in each JCL scan.
Subsystem	Use <b>JES2</b> , <b>JES3</b> , or <b>none</b> to indicate whether the JCL contains JES2 or JES3 statements to be validated for syntax. JES statements are not evaluated if <b>none</b> is selected for this option.
Suppress Error Messages	Select the type of messages to suppress in the result of a scan, including the JCL Scan view, the Problems view, and the JCL editor. The options are: <ul style="list-style-type: none"> <li>do not suppress</li> <li>advisory only</li> <li>advisory and warning</li> <li>advisory, warning, and error</li> <li>less than or equal to</li> </ul>
Severity Level	Select the severity level below or at which error messages are suppressed. This

Field Name	Description
Extended Error Text	<p>selection is enabled only if <b>less than or equal to</b> is selected for <b>Suppress Error Messages</b> option.</p> <p>Select the type of message text to be displayed for each error in the result of a scan. The options are:</p> <ul style="list-style-type: none"><li>• short message text</li><li>• extended message text</li></ul>

---

5. Click **Apply** to save the scan options or click **Apply and Close** to save the scan options and return to the **Connect to z/OS** dialog.

# Working with z/OS Datasets


This topic explains how to:

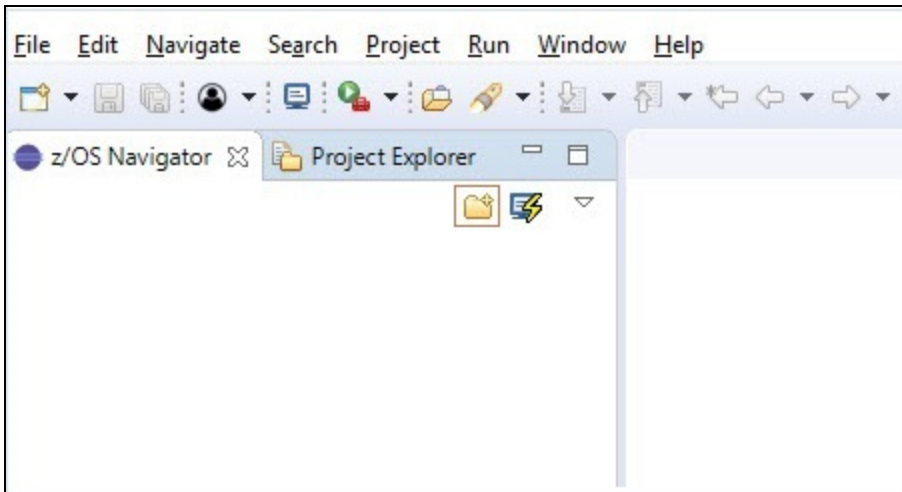
- display the z/OS datasets
- edit the z/OS datasets
- refresh the list of datasets

## Displaying Datasets

Datasets that are in the z/OS system can be viewed in z/OS Navigator or Host Explorer view.

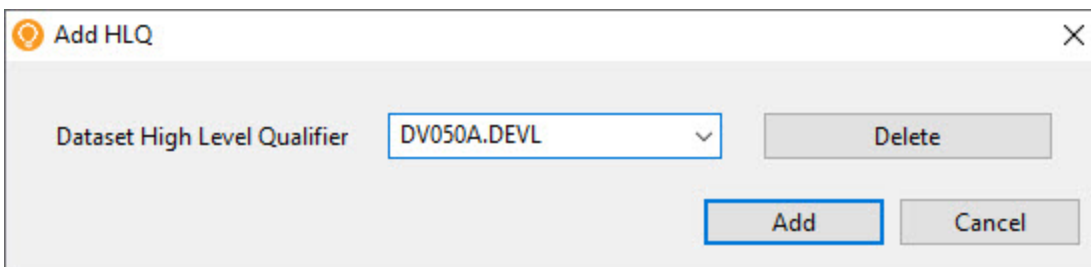
**To view datasets from z/OS Navigator:**

1. Select  (**Add High Level Qualifier**) icon from the **z/OS Navigator** pane. The **ADD HLQ** dialog box appears.



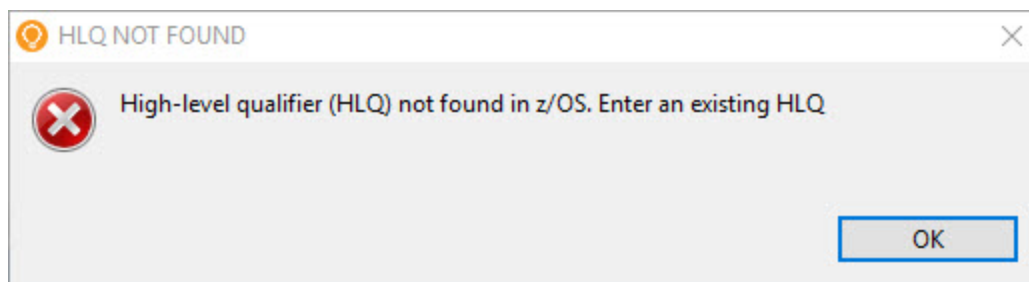
If connection to z/OS is not established, then **Connect to z/OS** dialog appears and prompts for main-frame credentials. See [Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator](#) for more information.

2. In the **Dataset High Level Qualifier**, enter the dataset's high-level qualifier that you would like to view and click **Add**. Datasets are displayed under a High-Level Qualifier (HLQ) in a tree-like structure.

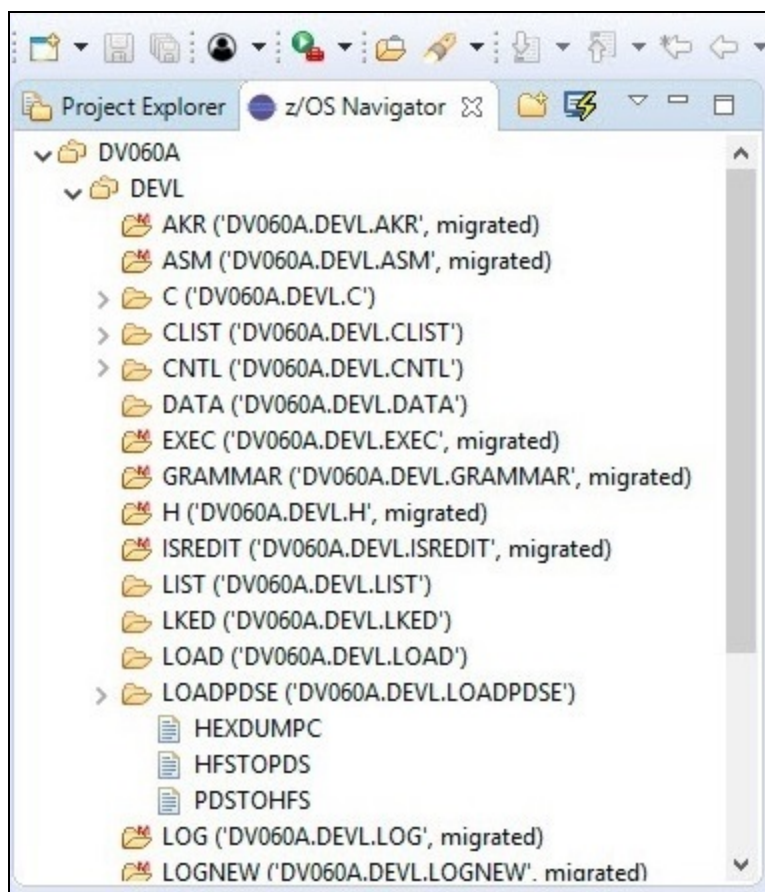




- If you have entered the high-level qualifier for the first time, then on clicking **Add**, the HLQ is saved to the **Dataset High Level Qualifier** list. The list keeps track of the HLQs that you have entered in the past so you don't have to retype again.
- To delete a saved HLQ from the **Dataset High Level Qualifier** list, select the HLQ and click **Delete**.
- If the entered HLQ is not found in z/OS, then the following error message is displayed. Make sure to enter an existing HLQ.



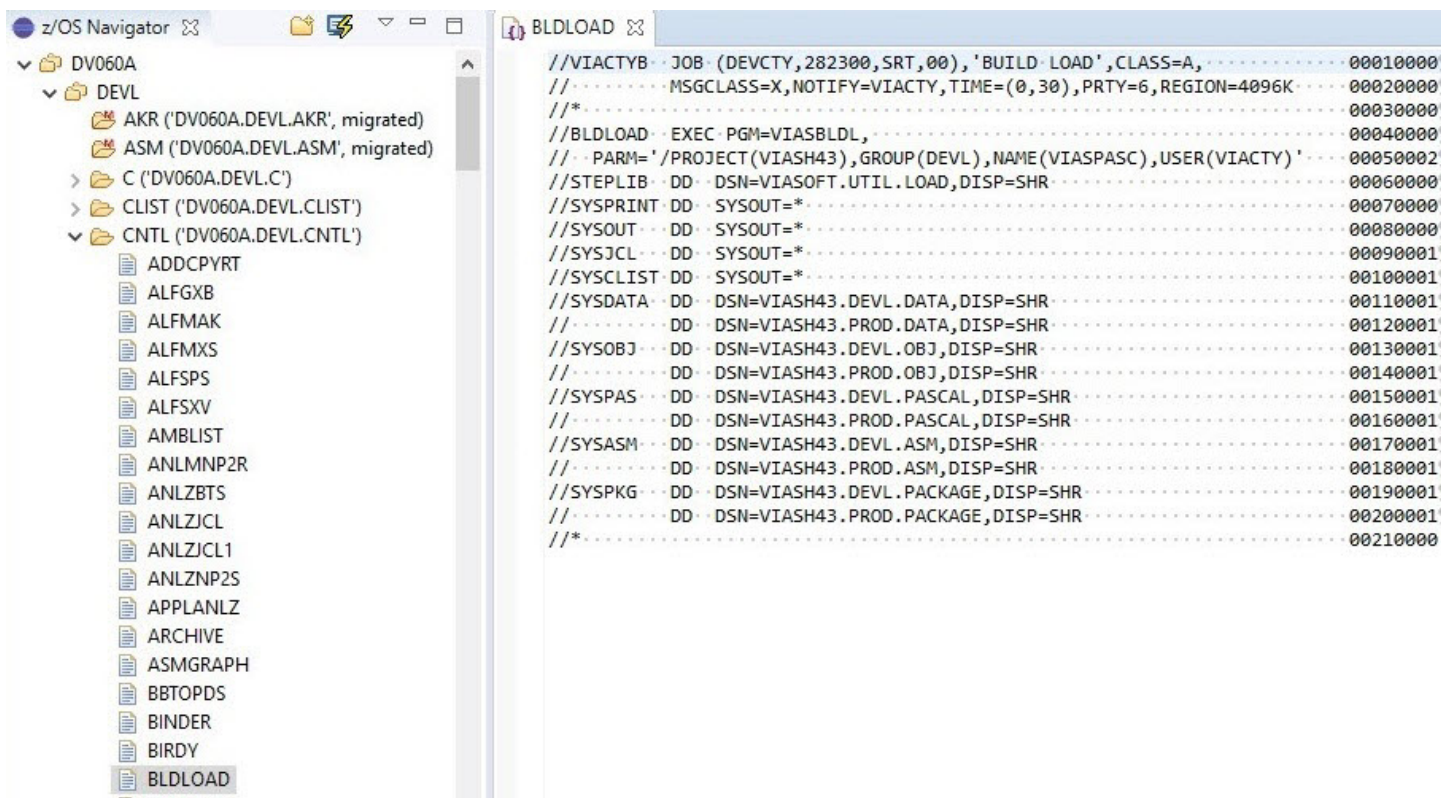
3. Expand the tree-like structure to view the files.



Icons in the tree structure represent the following:

Icon	Description
	node in a dataset qualifier
	PDS or PDSE containing dataset members
	sequential dataset
	partitioned dataset that has been migrated
	sequential dataset that has been migrated
	member in PDS or PDSE

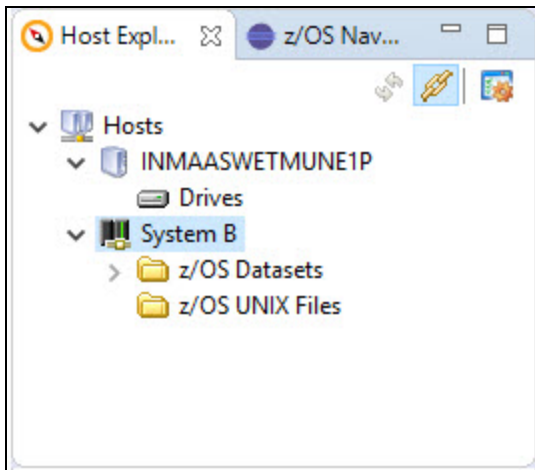
- Double-click on a dataset, the z/OS navigator opens the file in the editor. A tab in the editor displays the name of the file, while its contents appear in the main region of the tabbed pane.



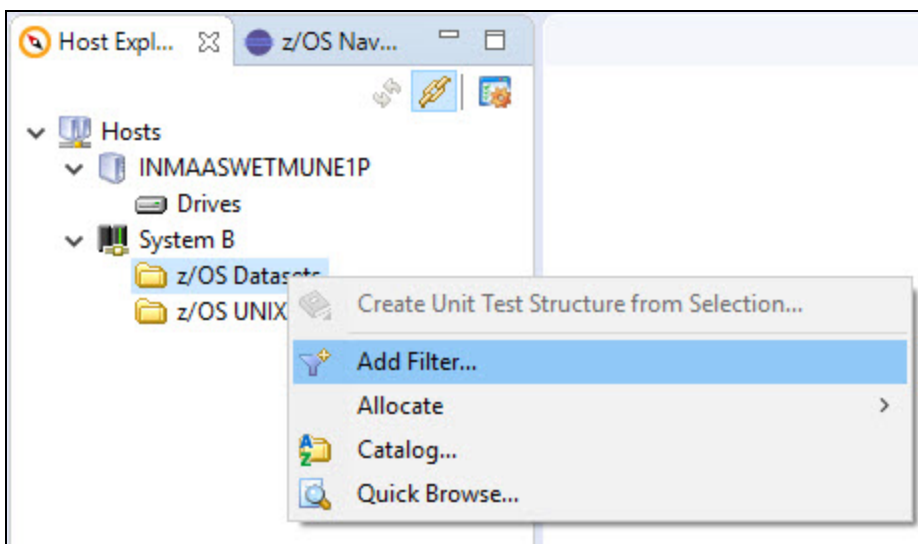
**To view datasets from Host Explorer:**

To view z/OS datasets using Host Explorer, you must define a z/OS connection in Host Explorer view. See [Connecting to z/OS from Host Explorer](#) for more information.

1. Expand the HCI connection in **Host Explorer**. Two folders namely **z/OS Datasets** and **z/OS UNIX Files** are displayed.

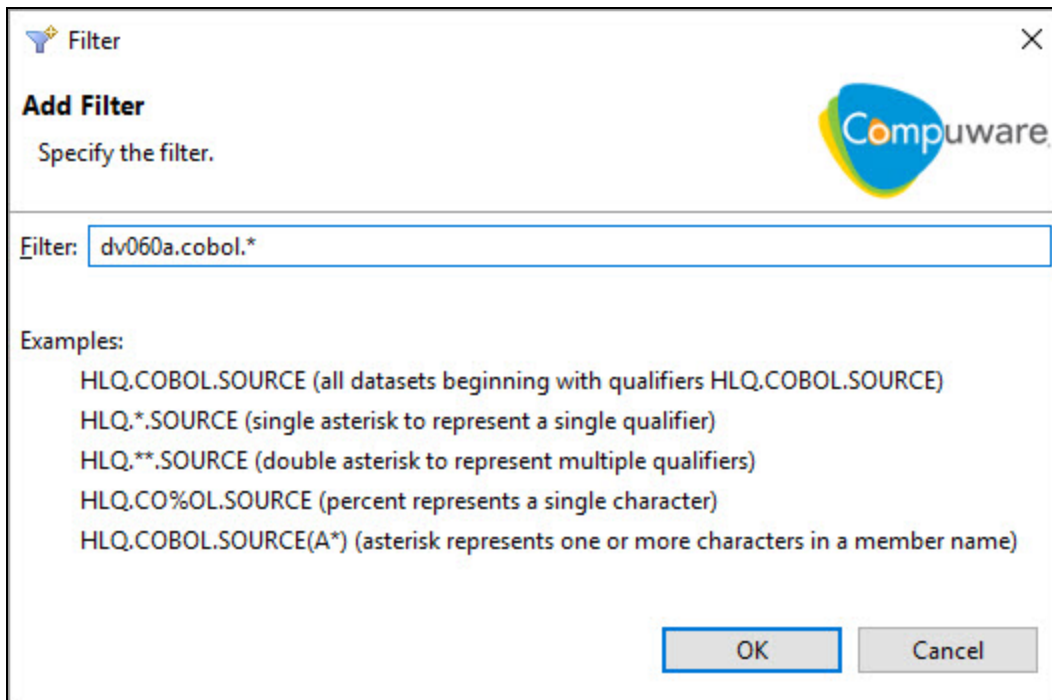


2. Right-click on the **z/OS Datasets** folder and select **Add Filter...**

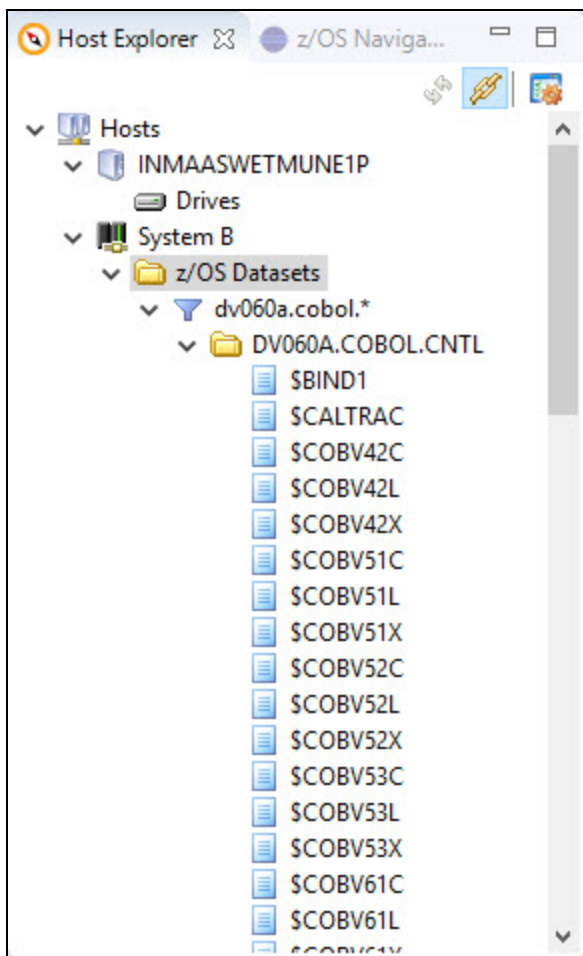


3. In the **Filter** dialog, enter a dataset pattern (fully qualified dataset name) for the z/OS datasets you want to access. You can use the asterisk (\*) wild card at any node of the dataset name including the first character.





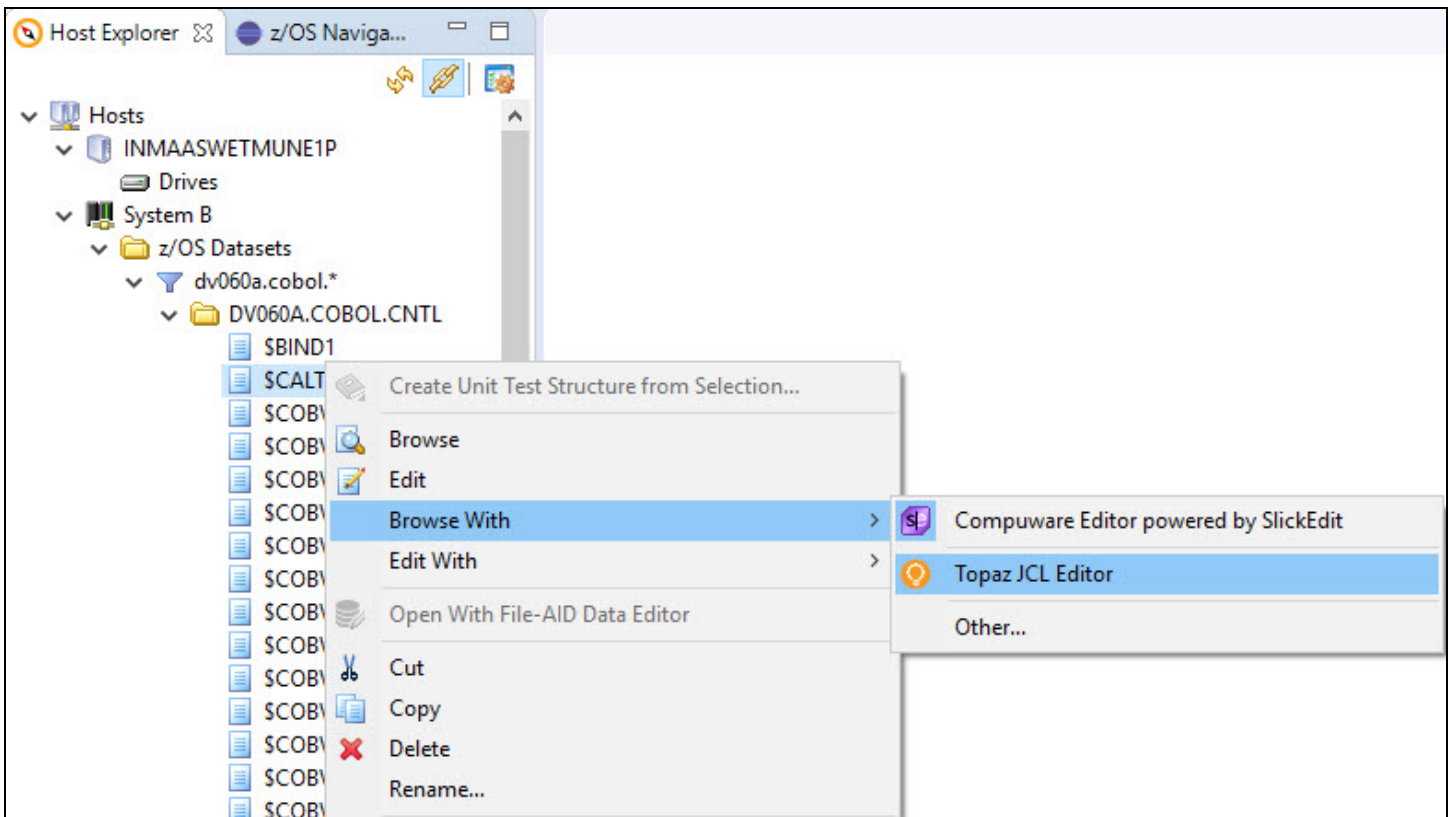
4. Click **OK**. Datasets are displayed in the Host Explorer under the specified filter.



5. Select the z/OS dataset you want to browse or edit, then right-click and select **Browse With --> Topaz JCL Editor** or **Edit With --> Topaz JCL Editor**.

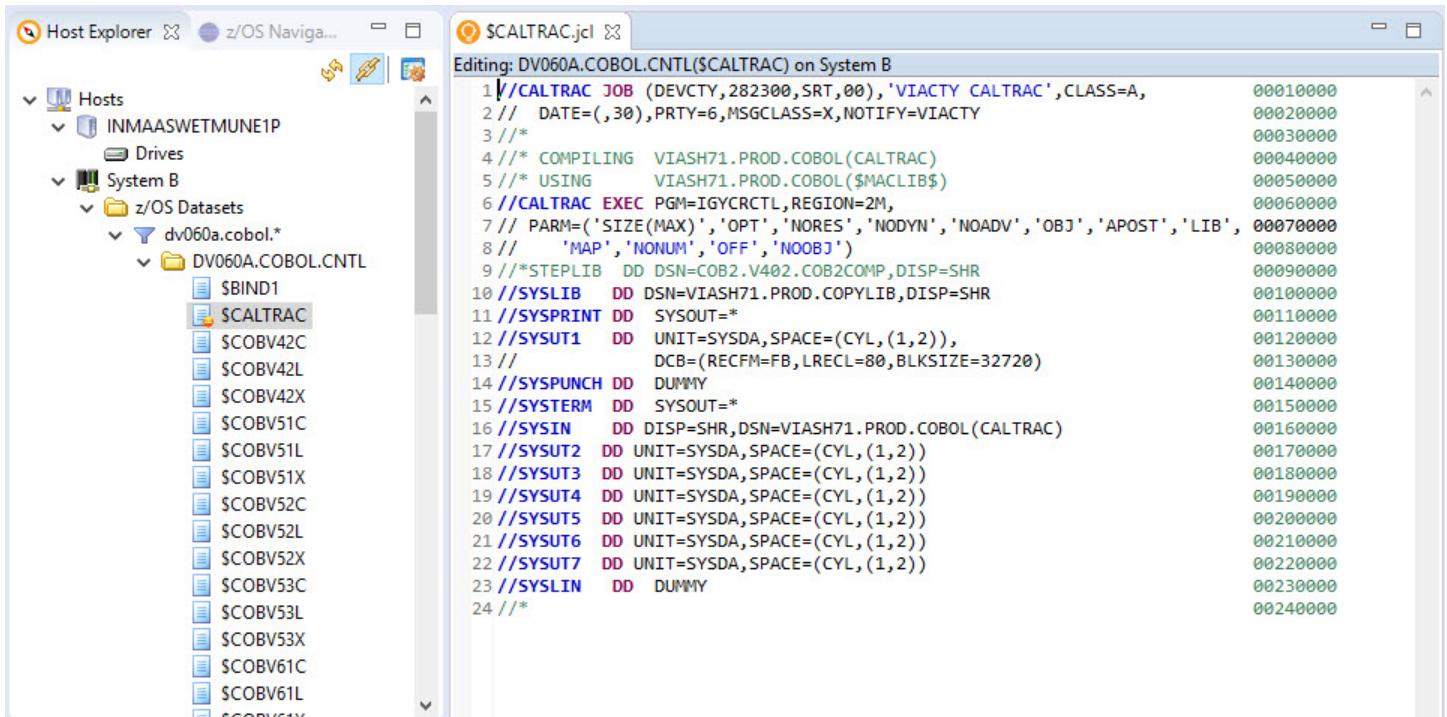


If you don't find the **Topaz JCL Editor** option, select **Other...**, and in the **Editor Selection** dialog, select **Topaz JCL Editor** from the **Internal Editors** list.



When a file is opened from Host Explorer view, JOB/SCAN commands are available in Topaz JCL Editor **only**. The commands are displayed as buttons in the toolbar as well as additional options in the Edit menu. If you are not already connected to a JOB/SCAN server when you run the JOB/SCAN commands, you will be asked to connect to JOB/SCAN first before the commands are executed. Perform the steps in [Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator](#) to connect to JOB/SCAN's server.

- The dataset is displayed in the JCL Editor. The **Browse With** option opens the selected file in the editor region for read-only viewing. If you want to make changes in the file, then select **Edit With** option.

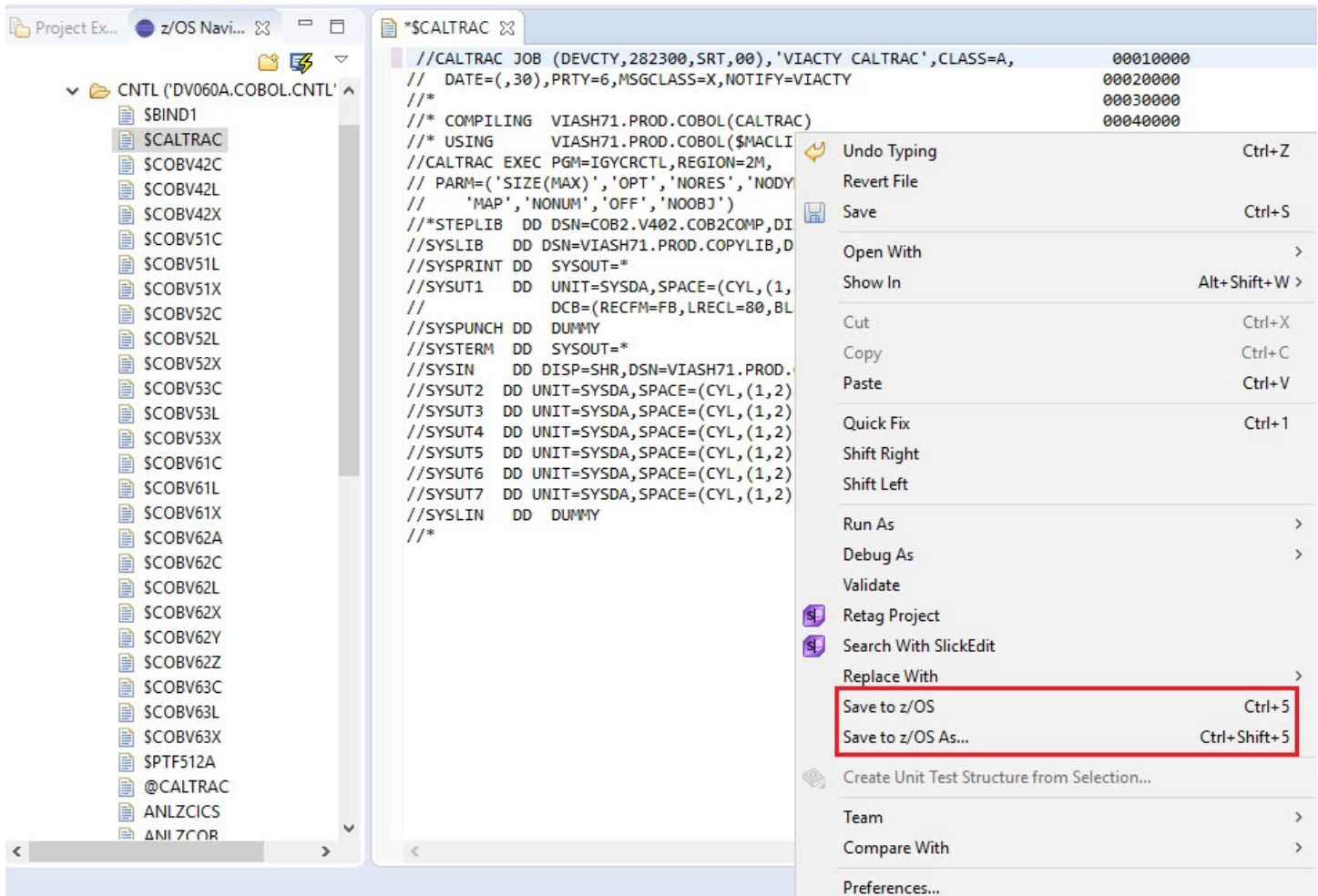


## Editing and Saving Datasets

The JCL editor allows the user to edit and save a dataset to z/OS.

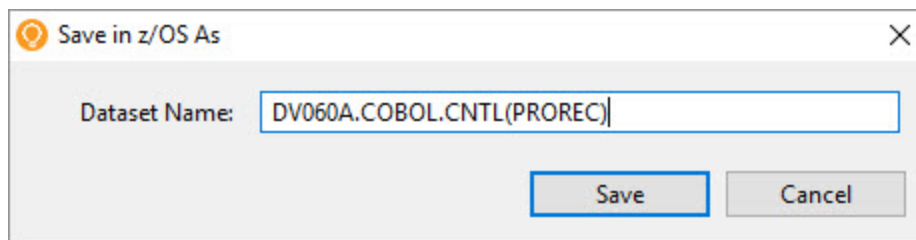
### **To edit and save a dataset using z/OS Navigator:**

1. In the **z/OS Navigator**, double-click on a dataset that you like to update. The dataset is displayed in the editor region.
2. After updating the dataset, right-click in the JCL editor area to bring up the dataset contextual menu.



3. Select one of the options to save the changes to z/OS server:

- a. **Save to z/OS** - This option allows the user to save the file under the same dataset name.
- b. **Save to z/OS As...** - This option allows the user to save the file either as a new member in a PDS or as an existing PS file. When this option is selected, user is prompted to enter a dataset name. Enter a name and click **OK**.



★ This option will not allocate a new dataset.

4. Refresh the z/OS Navigator to view the changes.

**To edit and save a dataset using Host Explorer:**

1. In the **Host Explorer**, right-click on a dataset that you like to update and select **Edit With --> Topaz JCL Editor**. The dataset is displayed in the **Topaz JCL Editor**.
2. After updating the file, do one of the following:
  - Press **Ctrl+ S**
  - Select **File > Save** or **Save As**
  - Right-click on the file and select **Save**.

When you save the changes, the dataset on the z/OS server is automatically updated.

3. You may have to refresh to view the changes.

## Refreshing Dataset

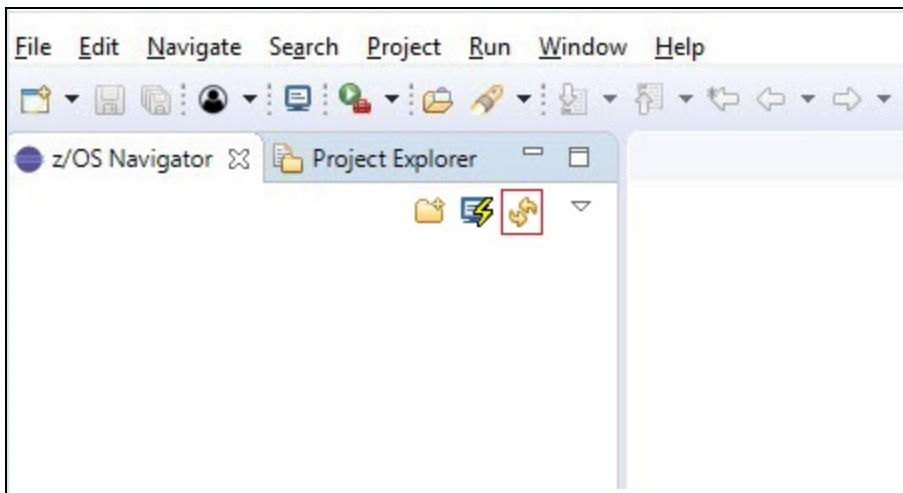
The z/OS Navigator and Host Explorer provides a refresh icon that refreshes the list of datasets.

You may need to refresh the list whenever you:

- save a file to z/OS
- add / delete / edit a dataset or a member in z/OS

**To refresh datasets:**

- Click the  icon from in the **z/OS Navigator** or **Host Explorer** view.



See [Validating JCL](#) for information on scanning / validating a file.

# Working with Local Files

ASG-JOB/SCAN Plug-in allows the user to scan files that are present in your local file system. This topic explains how to work with the files that are in your local file system.

## Displaying Local Files

Files in your local file system can be explored either from Host Explorer view or Project Explorer view.

### Display the files in Project Explorer

To display local files in the Project Explorer view, first they must be imported. You can import the files by one of the following methods:

- dragging and dropping from the file system
- copying and pasting from the file system
- using the Import option

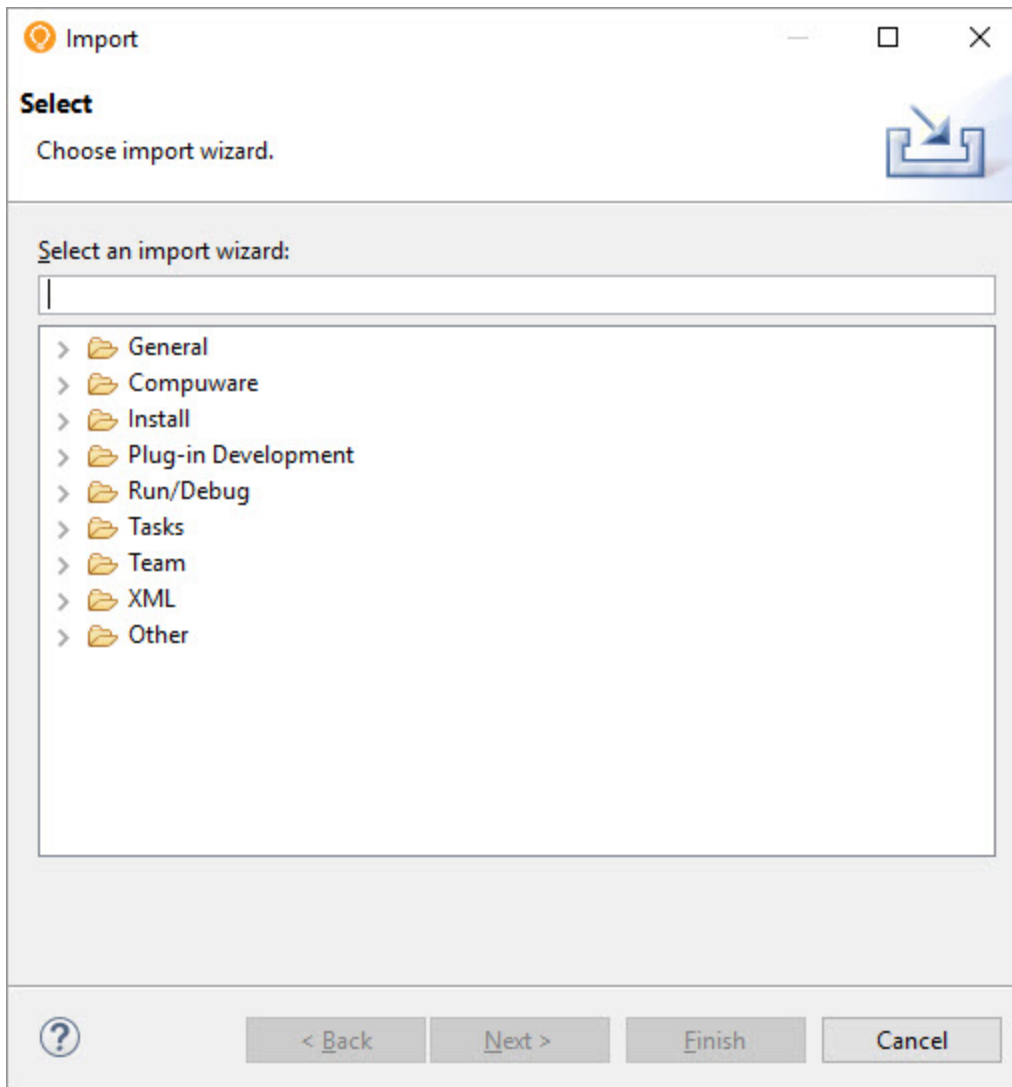
After importing the files, right-click on the file that you like to view and select **Open With --> Topaz JCL Editor**.



- If you don't find the **Topaz JCL Editor** option, select **Other...** and in the **Editor Selection** dialog, select **Topaz JCL Editor** from the **Internal Editors** list.
- When a file is opened from **Project Explorer** view, JOB/SCAN commands are available in Topaz JCL Editor **only**. The commands are displayed as buttons in the toolbar as well as additional options in the **Edit** menu. If you are not already connected to a JOB/SCAN server when you run the JOB/SCAN commands, you will be asked to connect to JOB/SCAN first before the commands are executed. Perform the steps in [Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator](#) to connect to JOB/SCAN's server.

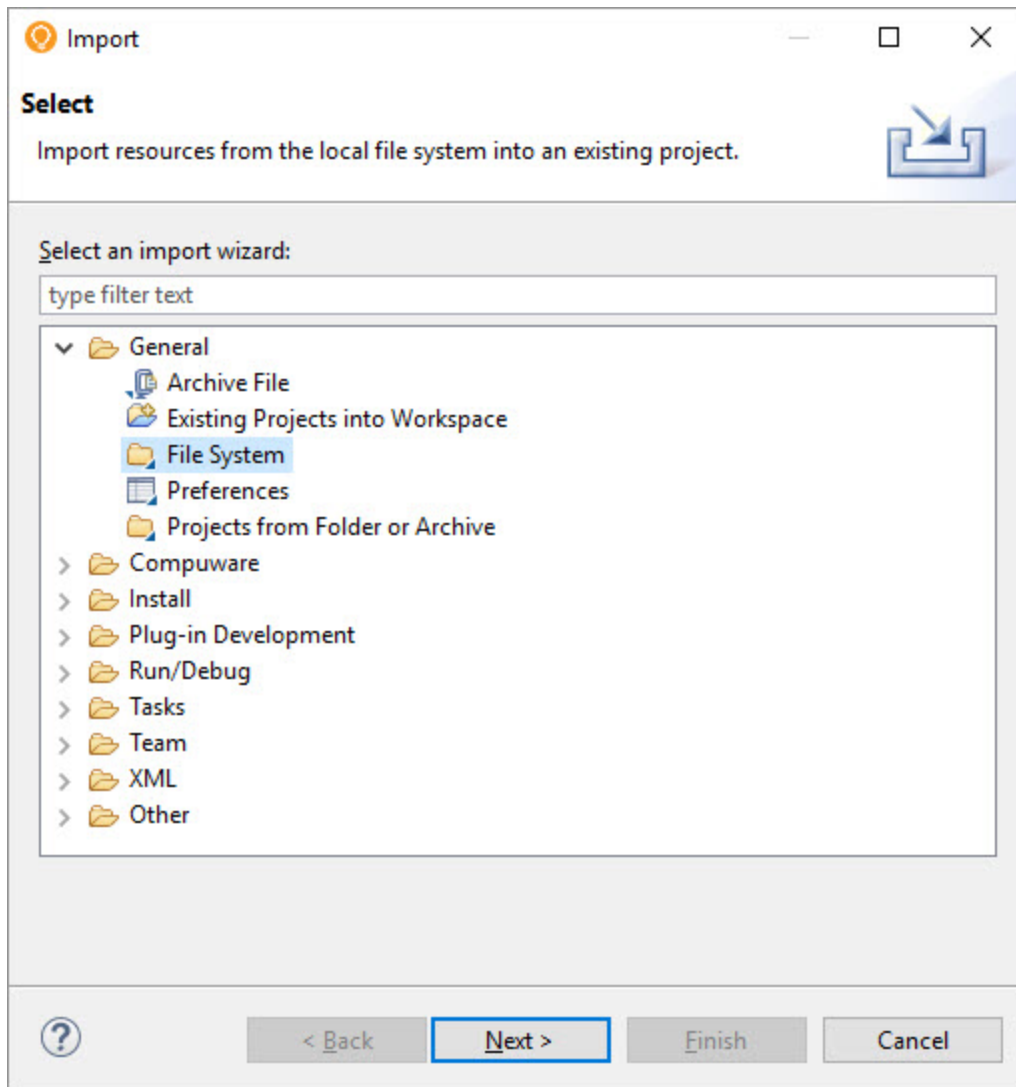
### To import files using Import option:

1. From the main menu, select **File > Import...** The **Import** dialog appears.

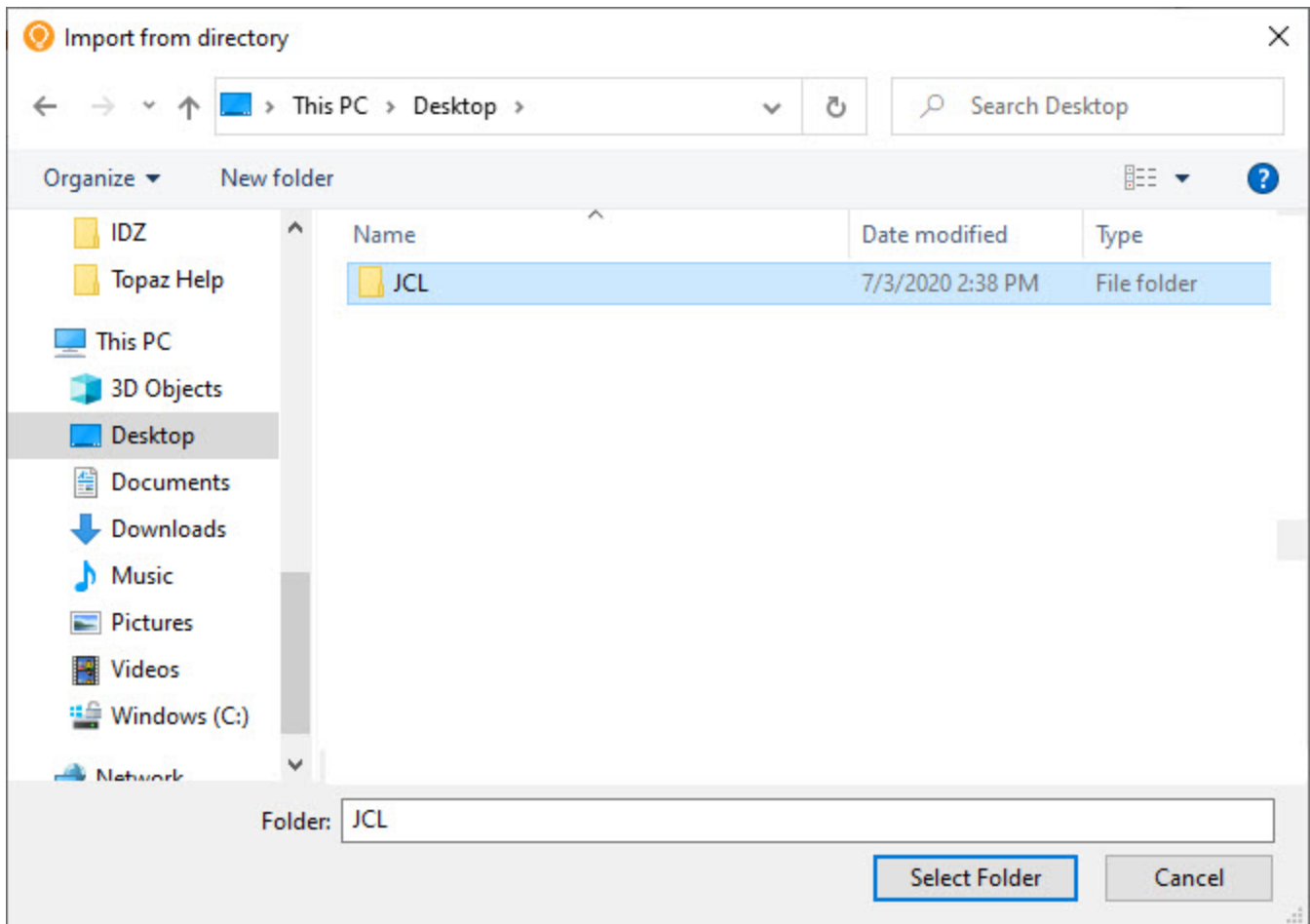


2. Expand the **General** folder, select **File System** and click **Next**.

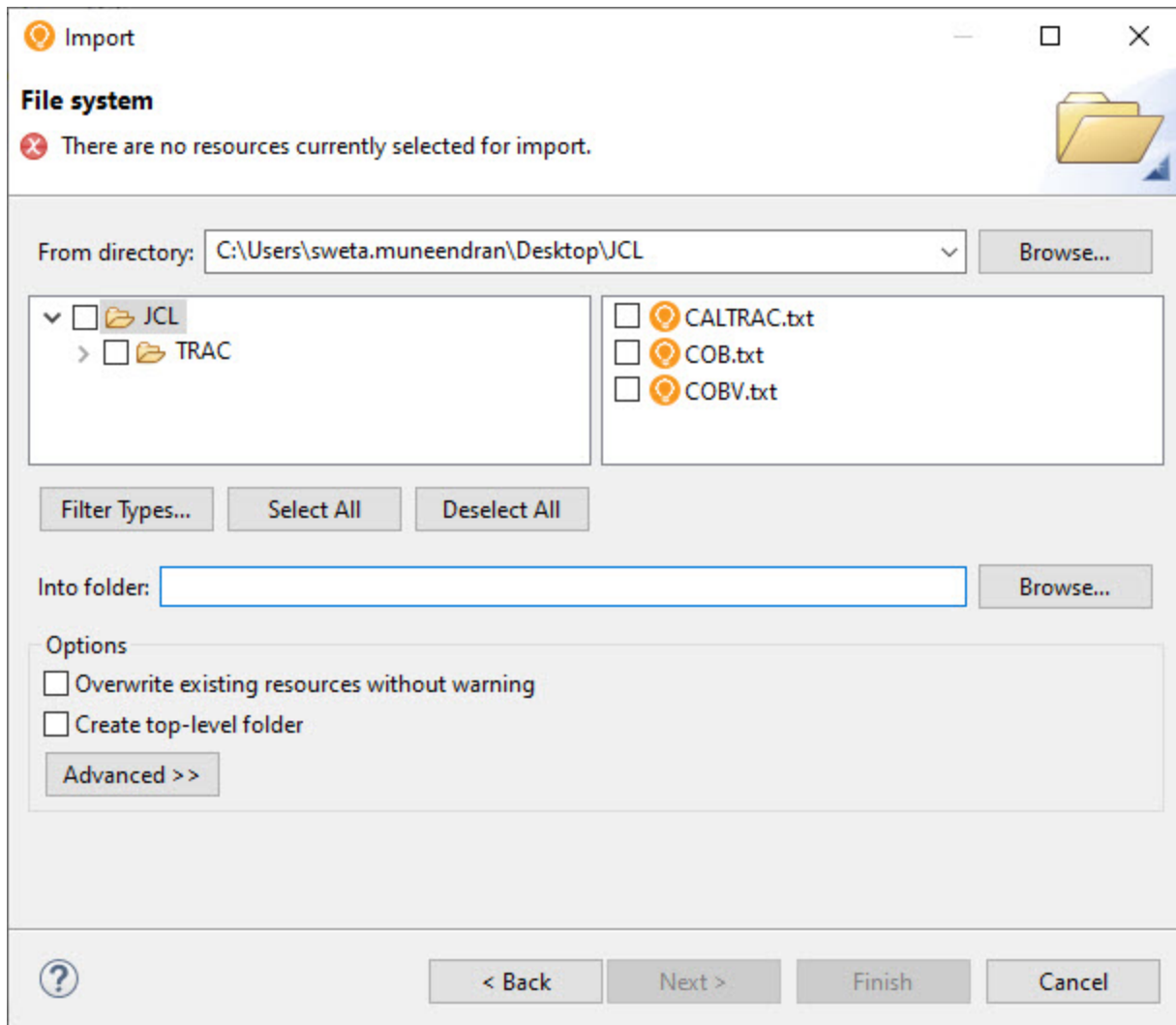




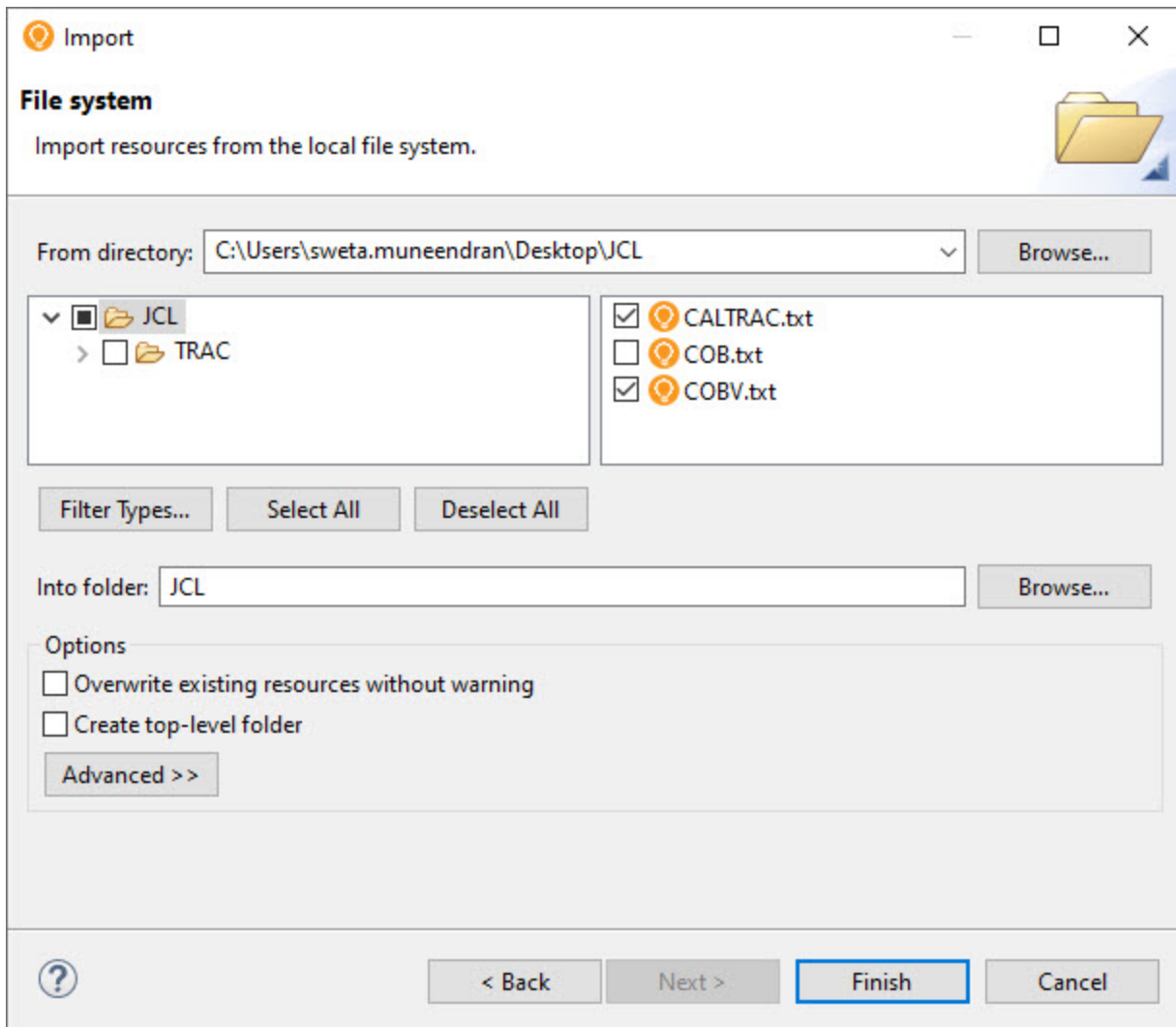
3. In the next page of the Import dialog, click the **Browse** button. The **Import from directory** dialog appears.



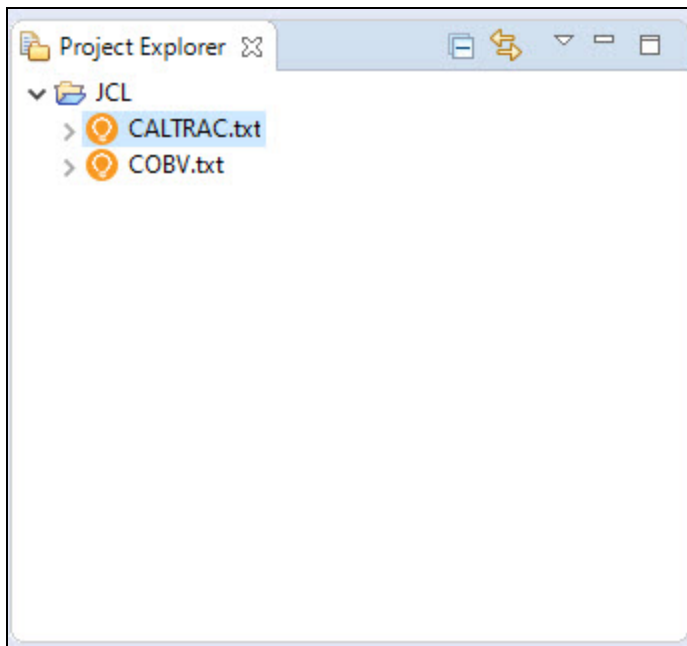
4. Select the directory from which you would like to add the files and click **Select Folder**. The directory is displayed in the left pane of the **Import** dialog.



5. To select exactly the files you want to add, expand the hierarchies in the left pane and select or clear the checkboxes that represent the folders in the selected directory. Then in the right pane, select or clear checkboxes for individual files.
6. In the **Into folder** option, specify the project or folder that will be the import destination.



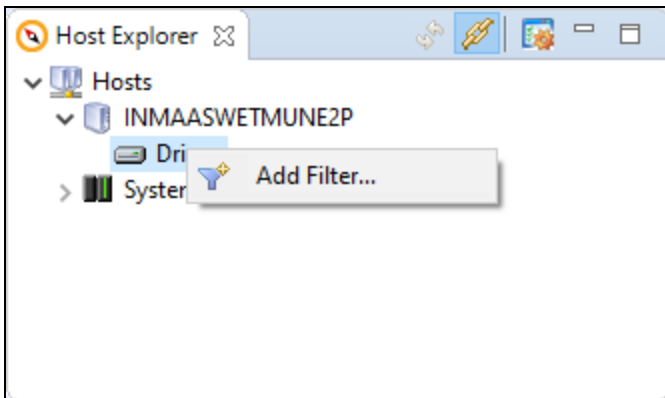
7. Click **Finish**. The files are displayed in the **Project Explorer** view under the specified destination.



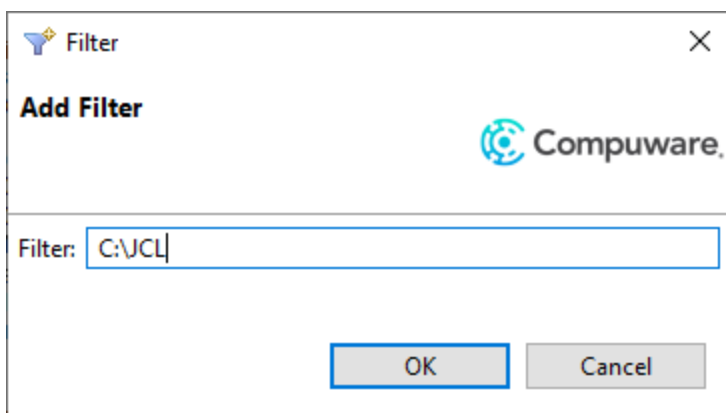
### Display files in Host Explorer

*To display the local files in Host Explorer, you need to add a local file system filter.*

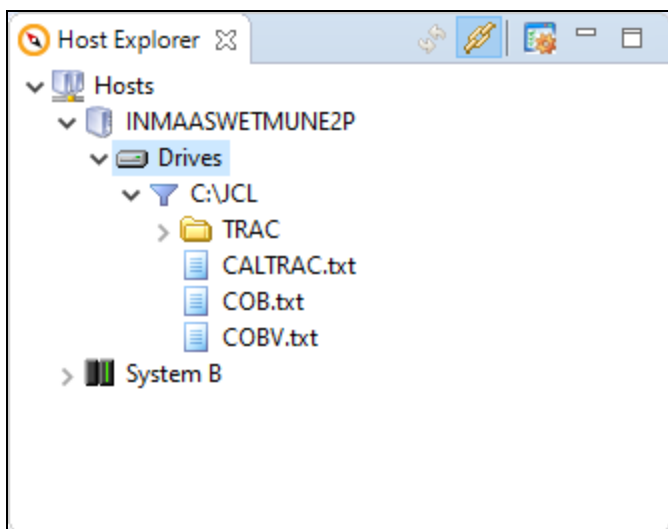
1. In the **Host Explorer** view, expand the machine name.



2. Right-click the resource type **Drives** and select **Add Filter**. The **Filter** dialog box appears.



3. Enter the filter, which can be a specific drive on the local machine or can include folder names (for example, C, D:\temp, or D:\temp\jcl).
4. Click **OK**, the results display all of the resources that match the filter.



5. Right-click on the file and select **Open With --> Topaz JCL Editor**.



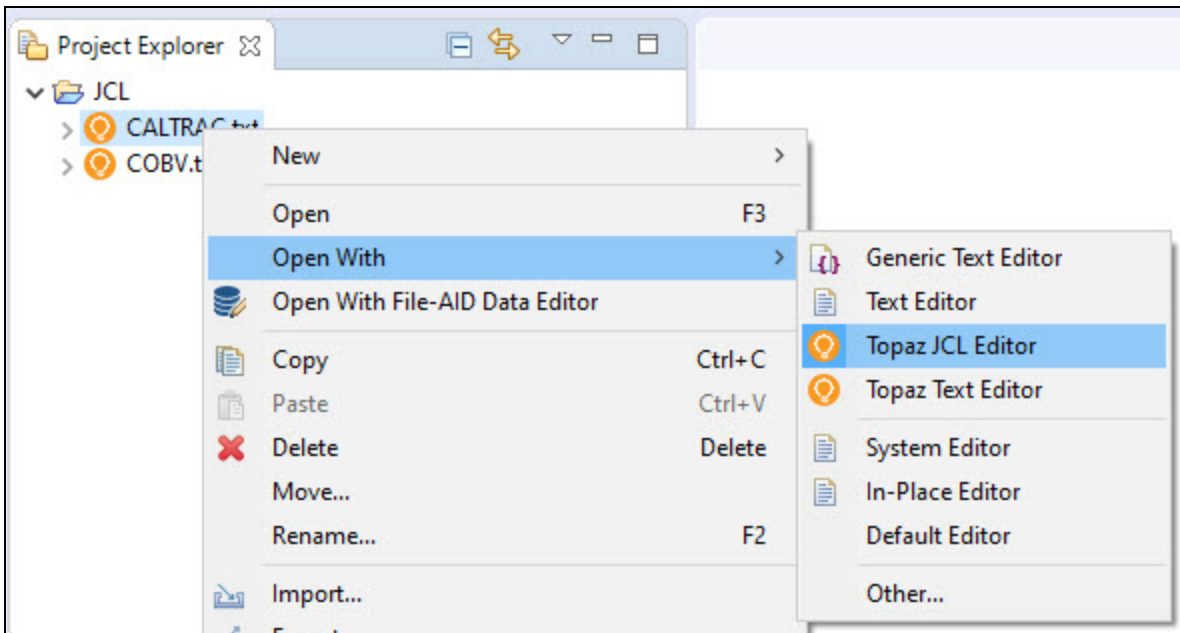
When a file is opened from **Host Explorer** view, JOB/SCAN commands are available in Topaz JCL Editor **only**. The commands are displayed as buttons in the toolbar as well as additional options in the Edit menu. If you are not already connected to a JOB/SCAN server when you run the JOB/SCAN commands, you will be asked to connect to JOB/SCAN first before the commands are executed. Perform the steps in [Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator](#) to connect to JOB/SCAN's server.

## Editing and Saving Local Files

The JCL editor allows you to edit and save a file that is in your local file system.

**To edit a file:**

1. In the **Project Explorer** view, right-click on the file that you want to edit and select **Open With > Topaz JCL Editor**. The file is opened in the JCL editor area.



If you don't find the **Topaz JCL Editor** option, select **Other...**, and in the **Editor Selection** dialog, select **Topaz JCL Editor** from the **Internal Editors** list.

2. After updating the file, do one of the following:
  - Press **Ctrl+ S**
  - Select **File > Save** or **Save As**
  - Right-click on the file and select **Save**.



- If the file opens in an external editor outside the Topaz workbench, then you probably opened the file without selecting an editor.
- ASG recommends you to open the file in Topaz JCL Editor.
- You may not see the JOB/SCAN functions if you have opened the file in different editor other than Topaz JCL Editor.

See [Validating JCL](#) for information on scanning / validating a file.

## Validating JCL

ASG-JOB/SCAN validates JCL in an input job stream, so that potential errors, including run-time, hidden, site-specific and syntax errors can be corrected before the job stream is submitted for processing. Similarly, ASG-JOB/SCAN Plug-in processes the entire job stream in the z/OS environment and displays the results in the JCL Scan and Problems views.

You can validate a JCL file using the following options:

- [Scan JCL](#)
- [Scan and submit JCL](#)


### Scan JCL

Scan JCL option available in the ASG-JOB/SCAN Plug-in is similar to ASG-JOB/SCAN's JEM and JSCAN edit macros. The edit macros validates the JCL file during an ISPF Edit session and displays the Structured JCL Listing (SJL) report. Similarly, the Scan JCL option validates the JCL file that is active in the JCL Editor and displays the results in JCL Scan and Problems views.

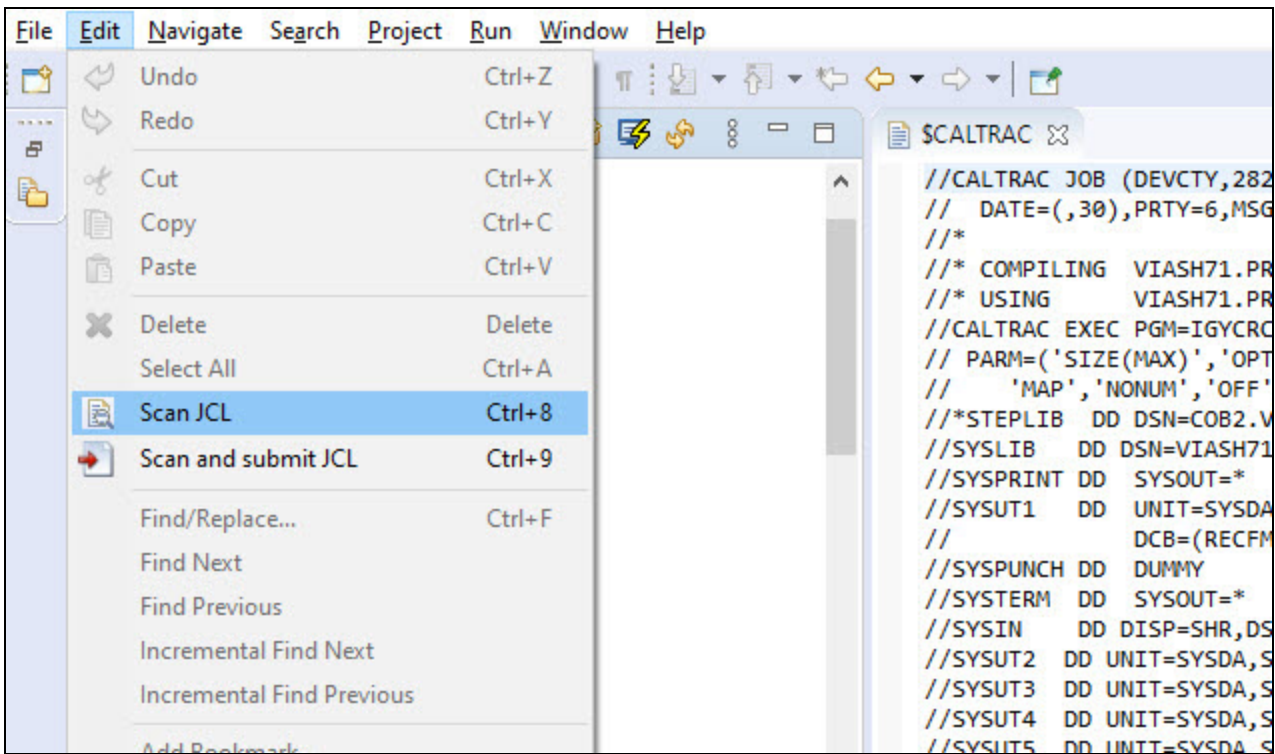


- If you are not connected to a JOB/SCAN server when you run the JOB/SCAN commands, you will be asked to connect to JOB/SCAN first before the commands are executed. Perform the steps in [Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator](#) to connect to JOB/SCAN's server.
- When a file is opened from **Host Explorer** or **Project Explorer** view, JOB/SCAN commands are available in Topaz JCL Editor **only**.
- JCL Scan Options and Processing Options must be specified before you begin to scan a JCL file. For more information, see [Setting the JCL Scan Options](#) and [Setting the Processing Options](#).

#### **To scan JCL:**

1. Open the file that you want to scan in JCL editor.
2. Make changes, if any, and save it.
3. Select **Scan JCL** option from the **Edit** menu or  icon from the toolbar.





4. JOB/SCAN Plug-in validates the input JCL and displays the reports in the **JCL Scan** and **Problems** views. For more information, refer to [Reports](#).



When you scan another file in the JCL editor, the report in the JCL Scan view is replaced. However, the messages in the Problems view are not replaced, instead they add up to the existing messages.




## Scan and submit JCL

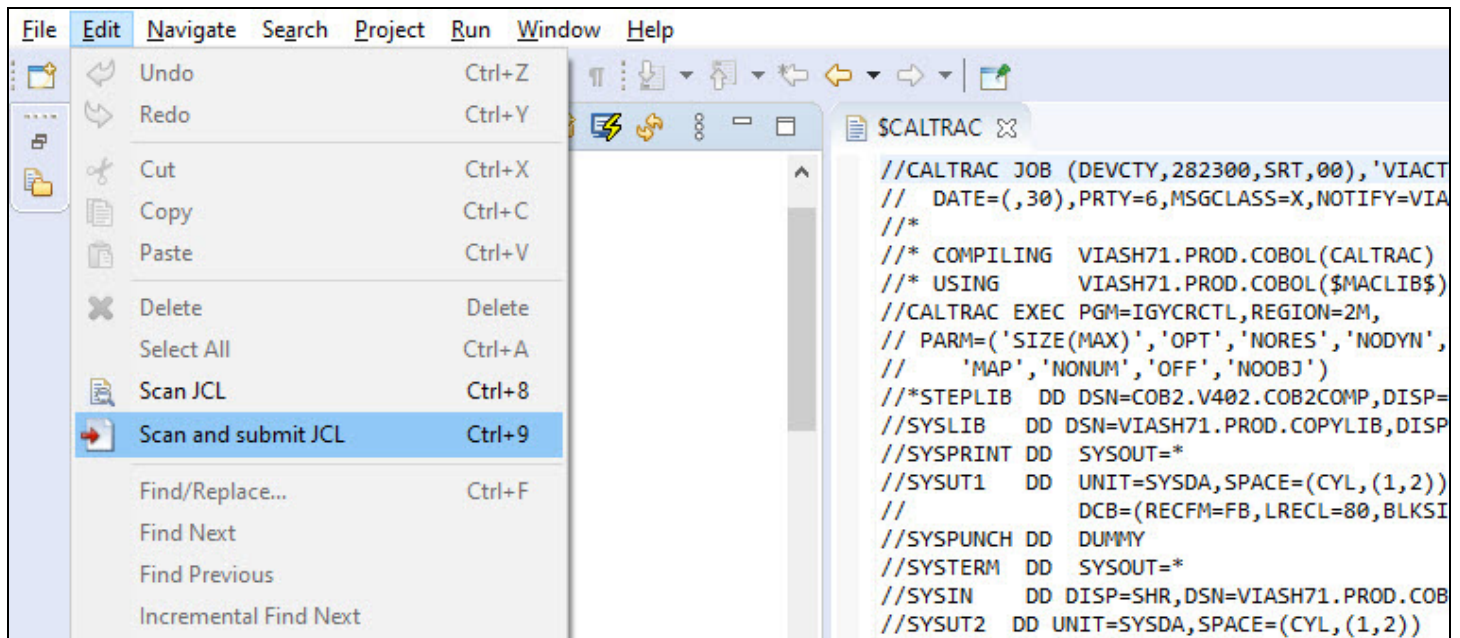
Scan and submit JCL option available in the ASG-JOB/SCAN Plug-in is similar to the ASG-JOB/SCAN's JSUB edit macro. The JJSUB edit macro validates a JCL and submits the job from the edit session. Similarly, the 'Scan and submit JCL' option validates a file that is active in the JCL Editor and submits the job to z/OS. If the validation process completes with a severity level of 4 or less, the job is automatically submitted and no reports are displayed in the JCL Scan or Problems views. If the severity level is greater than 4, then the job is not submitted and the reports are displayed.



- If you are not connected to a JOB/SCAN server when you run the JOB/SCAN commands, you will be asked to connect to JOB/SCAN first before the commands are executed. Perform the steps in [Connecting to z/OS and ASG-JOB/SCAN from z/OS Navigator](#) to connect to JOB/SCAN's server.
- When a file is opened from **Host Explorer** or **Project Explorer** view, JOB/SCAN commands are available in Topaz JCL Editor **only**.
- JCL Scan Options and Processing Options must be specified before you begin to scan a JCL file. For more information, see [Setting the JCL Scan Options](#) and [Setting the Processing Options](#).

### To scan and submit JCL:

1. Open the file that you want to scan in JCL Editor.
2. Make changes, if any, and save it.
3. Select **Scan and submit JCL** option from the **Edit** menu or  icon from the toolbar.



4. JOB/SCAN Plug-in validates the input JCL and displays the reports in the JCL Scan and Problems views depending on the severity. The job will be submitted if the severity is less than or equal to 4.

### Sample JCL file with severity level less than or equal to 4

The below sample JCL file with severity level less than or equal to 4 is automatically submitted to z/OS and a success message is displayed in the JCL Scan and JOB/SCAN Console views.

The screenshot displays the z/OS Navigator interface. On the left, a file tree shows the directory structure under 'CNTL ('DV060A.COBOL.CNTL')', with files like SBIND1, SCALTRAC, and SCOBV42C. The main window shows the JCL file 'SCOBV42C' with the following content:

```
//COBV42C JOB (DEVCTY,282300,SRT,00),'COBOL 4.2 COMP',CLASS=A,PRTY=6, 00010000
//      NOTIFY=&SYSUID,MSGCLASS=X,REGION=0M,TIME=20      00020000
//*      00030000
//COBCOMP PROC SRC='DV060A.NTIS',      00040000
//      HLQ='DV060A.COBV42',      00050000
//      MEM=      00060000
//STEPCOMP EXEC PGM=IGYCRCTL,REGION=0M,      00070000
//      PARM='NOOPT,LIST,MAP,NUM,NOOFF,S'      00080000
//* PARM='LIST,MAP,S,NOADV,FASTSRT,RENT,TRUNC(OPT),NOXREF'      00090000
//*TEPLIB DD DSN=COB2.R32.COB2COMP,DISP=SHR      00100000
//*TEPLIB DD DSN=SYSP.IGY122.SIGYCOMP,DISP=SHR      00110000
//*TEPLIB DD DSN=ESW.MVSCOB.IGY210.SIGYCOMP,DISP=SHR      00120000
//*TEPLIB DD DSN=SYSP.IGY331.SIGYCOMP,DISP=SHR      00130000
//*TEPLIB DD DSN=SYSP.IGY340.SIGYCOMP,DISP=SHR      00140000
//*TEPLIB DD DSN=SYSP.IGY410.SIGYCOMP,DISP=SHR      00150000
//STEPLIB DD DSN=SYSP.IGY420.SIGYCOMP,DISP=SHR,DCB=BLKSIZE=32760      00160000
//      DD DSN=CEE.SCEERUN,DISP=SHR      00170000
//      DD DSN=CEE.SCEERUN2,DISP=SHR      00180000
//SYSIN DD DSN=&SRC..COBOL(&MEM),DISP=SHR      00190000
//SYSLIB DD DSN=&SRC..COBOL,DISP=SHR      00200000
```

Below the JCL file, the 'JCL Scan' view shows the message: 'Job submitted to z/OS.' The 'Console' view shows the following output:

```
JOB/SCAN Console
JCL severity level is 0. Job is ready to be submitted to z/OS.
Job submitted to z/OS successfully.
```

The status bar at the bottom indicates '229M of 687M'.

### Sample JCL file with severity level greater than 4

The below sample JCL file with severity level greater than 4 is not submitted and a failure message along with severity level is displayed in the JOB/SCAN Console view. Validation reports are displayed in JCL Scan and Problems views. For more information on reports, refer to [Reports](#).

The screenshot displays the z/OS Navigator interface with the following components:

- z/OS Navigator:** Shows a tree view of files under 'CNTL (DV060A.COBOL.CNTL)', including 'SCALTRAC' and several 'SCOBV' files.
- JCL Editor:** Displays the JCL code for 'SCALTRAC'. Key lines include:
 

```
//CALTRAC JOB (DEVCTY,282300,SRT,00),'VIACTY CALTRAC',CLASS=A,
//DATE=(,30),PRTY=6,MSGCLASS=X,NOTIFY=VIACTY
//**
//** COMPILING VIASH71.PROD.COBOL(CALTRAC)
//** USING VIASH71.PROD.COBOL($MACLIB$)
//CALTRAC EXEC PGM=IGYCRCTL,REGION=2M,
// PARM=('SIZE(MAX)', 'OPT', 'NORES', 'NODYN', 'NOADV', 'OBJ', 'APOST', 'LIB',
// 'MAP', 'NONUM', 'OFF', 'NOOBJ')
//**STEPLIB DD DSN=COB2.V402.COB2COMP,DISP=SHR
//SYSLIB DD DSN=VIASH71.PROD.COPYLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,2)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=32720)
//SYSPPUNCH DD DUMMY
//SYSTEM DD SYSOUT=*
//SYSIN DD DISP=SHR,DSN=VIASH71.PROD.COBOL(CALTRAC)
//SYSUT2 DD UNIT=SYSDA,SPACE=(CYL,(1,2))
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,2))
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(1,2))
//SYSUT5 DD UNIT=SYSDA,SPACE=(CYL,(1,2))
//SYSUT6 DD UNIT=SYSDA,SPACE=(CYL,(1,2))
//SYSUT7 DD UNIT=SYSDA,SPACE=(CYL,(1,2))
//SYSTN DD NIMMY
```
- Problems View:** Shows '2 errors, 0 warnings, 1 other'. It lists 'Errors (2 items)' and 'Infos (1 item)'. The error details are partially visible: '\*\*\*ERROR - D5S1080E - INVALID JOB PARAMETER - "DATE "'.
- JCL Scan View:** Shows the same error message as the Problems view: '\*\*\*ERROR - D5S1080E - INVALID JOB PARAMETER - "DATE "'.
- Console View:** Shows system messages:
 

```
LOGGING IN TO z/OS AT mvssysb.asg.com:32208, User: dv423a
Logged in to z/OS on 2020/08/31 12:01:19.
Connecting to JOB/SCAN REST Services at mvssysb.asg.com:62000, User: dv423a
Connected to ASG-JOB/SCAN.
JCL severity level is 8. Job is not ready to be submitted to z/OS.
```



When you scan another file in the JCL editor, the report in the JCL Scan view is replaced. However, the messages in the Problems view are not replaced, instead they add up to the existing messages.

## Reports

ASG-JOB/SCAN Plug-in validates the input JCL for any errors, warnings and advisories (informational). The results of the validation are displayed in an easy-to-read report format. These reports are displayed automatically to the user in the **JCL Scan** and **Problems** views.

The reports contain the following messages:

### Error

These messages are likely to affect processing and should be corrected before the JCL is submitted.

### Warning

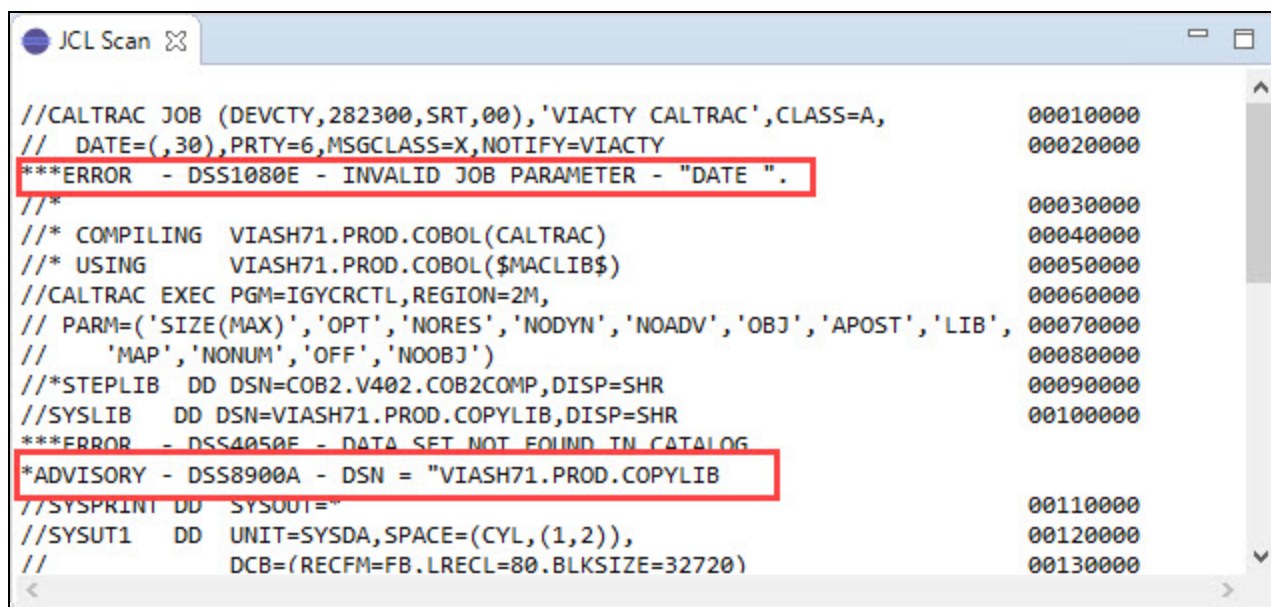
These messages may or may not affect processing and should be examined to determine if they must be corrected before the JCL is submitted.

### Advisory

These messages are informational only and are unlikely to affect processing.

## JCL Scan Report

In the JCL Scan report, the messages appear immediately after the JCL line that contains errors or warnings or advisories.



```

JCL Scan
//CALTRAC JOB (DEVCTY,282300,SRT,00),'VIACTY CALTRAC',CLASS=A,          00010000
// DATE=(,30),PRTY=6,MSGCLASS=X,NOTIFY=VIACTY                       00020000
***ERROR - DSS1080E - INVALID JOB PARAMETER - "DATE ".
//*                                                                    00030000
//* COMPILING VIASH71.PROD.COBOBOL(CALTRAC)                          00040000
//* USING      VIASH71.PROD.COBOBOL($MACLIB$)                        00050000
//CALTRAC EXEC PGM=IGYCRCTL,REGION=2M,                                00060000
// PARM=('SIZE(MAX)', 'OPT', 'NORES', 'NODYN', 'NOADV', 'OBJ', 'APOST', 'LIB', 00070000
//   'MAP', 'NONUM', 'OFF', 'NOOBJ')                                  00080000
//*STEPLIB DD DSN=COB2.V402.COBOB2COMP,DISP=SHR                       00090000
//SYSLIB DD DSN=VIASH71.PROD.COPYLIB,DISP=SHR                        00100000
***ERROR - DSS4050E - DATA SET NOT FOUND IN CATALOG
*ADVISORY - DSS8900A - DSN = "VIASH71.PROD.COPYLIB
//SYSPRINT DD SYSOUT=*                                              00110000
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,2)),                            00120000
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=32720)                             00130000

```

The messages are displayed in following format:

[Message Number] [Message Type] - [Message Text]

- Message types are:
  - Advisory
  - Error

- Warning
- Message number is a five digit numeral prefixed with DSS and suffixed with message type.
- Message text is a brief description of the problem.

After the messages, the report summarizes the Statistical Summary and the Space Analysis.

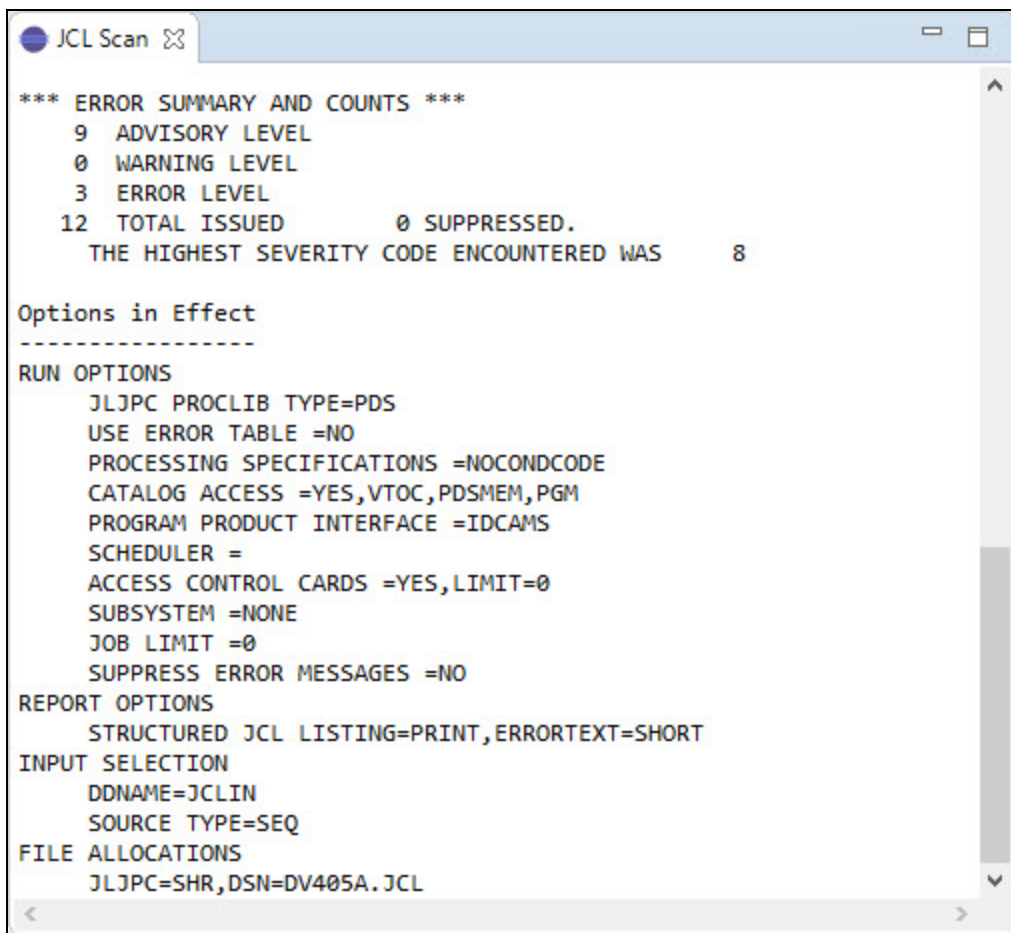
The Statistical Summary includes the total number of PROGRAM(S), PROC(S), INCLUDE(S), DD STATEMENT(S) and input / output TAPE DD STATEMENT(S) that are in the scanned JCL file.

```

JCL Scan
*ADVISORY - DSS8940A - TEMPORARY DATA SET WILL BE USED.
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,2))                00180000
*ADVISORY - DSS8940A - TEMPORARY DATA SET WILL BE USED.
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(1,2))                00190000
*ADVISORY - DSS8940A - TEMPORARY DATA SET WILL BE USED.
//SYSUT5 DD UNIT=SYSDA,SPACE=(CYL,(1,2))                00200000
*ADVISORY - DSS8940A - TEMPORARY DATA SET WILL BE USED.
//SYSUT6 DD UNIT=SYSDA,SPACE=(CYL,(1,2))                00210000
*ADVISORY - DSS8940A - TEMPORARY DATA SET WILL BE USED.
//SYSUT7 DD UNIT=SYSDA,SPACE=(CYL,(1,2))                00220000
*ADVISORY - DSS8940A - TEMPORARY DATA SET WILL BE USED.
//SYSLIN DD DUMMY                                       00230000
//*                                                       00240000
***ERROR - DSS2050E - PROGRAM "TGYCRCTI" NOT FOUND
JOB (CALTRAC ) CONSISTS OF      0 PROCS,      1 PROGRAMS,      13 DD STATEMENTS, AND      24 TOTAL RECORDS.
      THERE ARE      0 INPUT AND      0 OUTPUT      TAPE DD STATEMENTS.
      SPACE ANALYSIS      (DISK ) TEMP CYL      7 -PRIMARY      14 -SEC
  
```

The Space Analysis includes list of all new space requirements for the validated job stream. The total allocation requirements are broken down by UNIT and VOLSER, and indicate whether a file is permanent or temporary.

Next, the ERROR SUMMARY AND COUNTS section summarizes the total number of errors, warnings, and advisories encountered after scanning the JCL. This section is displayed only when you select the **Include summary in JCL scan output** option in the **JCL Scan Options** tab in the **JOB/SCAN** preference page.



```

JCL Scan
*** ERROR SUMMARY AND COUNTS ***
 9 ADVISORY LEVEL
 0 WARNING LEVEL
 3 ERROR LEVEL
12 TOTAL ISSUED      0 SUPPRESSED.
  THE HIGHEST SEVERITY CODE ENCOUNTERED WAS      8

Options in Effect
-----
RUN OPTIONS
  JLJPC PROCLIB TYPE=PDS
  USE ERROR TABLE =NO
  PROCESSING SPECIFICATIONS =NOCONDCODE
  CATALOG ACCESS =YES,VTOC,PDSMEM,PGM
  PROGRAM PRODUCT INTERFACE =IDCAMS
  SCHEDULER =
  ACCESS CONTROL CARDS =YES,LIMIT=0
  SUBSYSTEM =NONE
  JOB LIMIT =0
  SUPPRESS ERROR MESSAGES =NO
REPORT OPTIONS
  STRUCTURED JCL LISTING=PRINT,ERRORTXT=SHORT
INPUT SELECTION
  DDNAME=JCLIN
  SOURCE TYPE=SEQ
FILE ALLOCATIONS
  JLJPC=SHR,DSN=DV405A.JCL

```

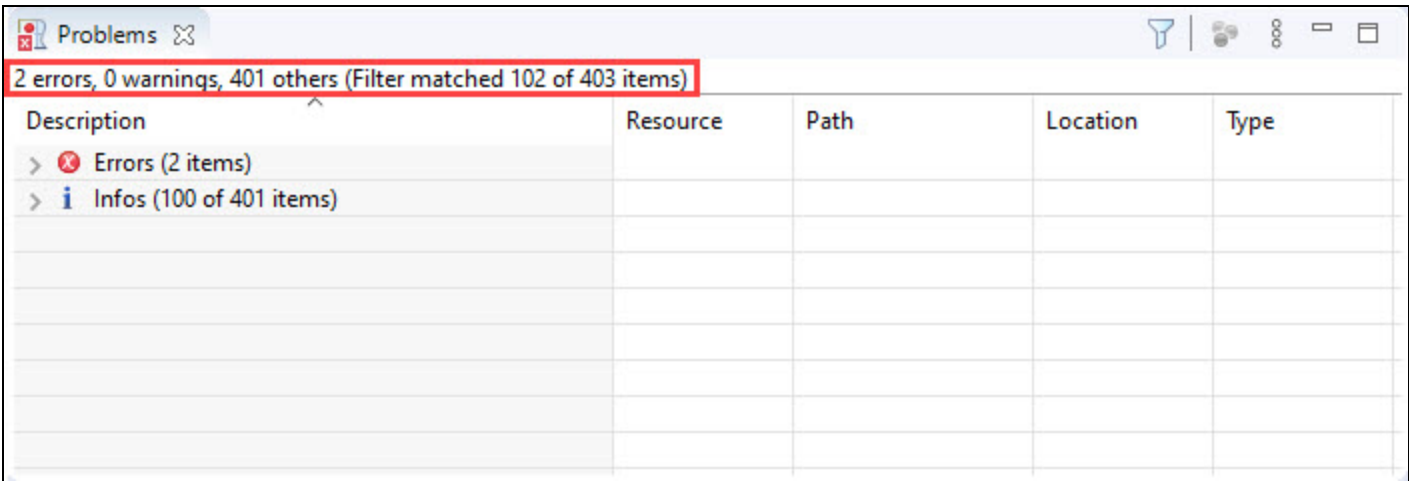
At the end is the Options in Effect section. This section summarizes the JCL scan options and the processing options that are used in validating a JCL. This section is displayed only when you select the **Include summary in JCL scan output** option in the **JCL Scan Options** tab in the **JOB/SCAN** preference page.

## Problems Report

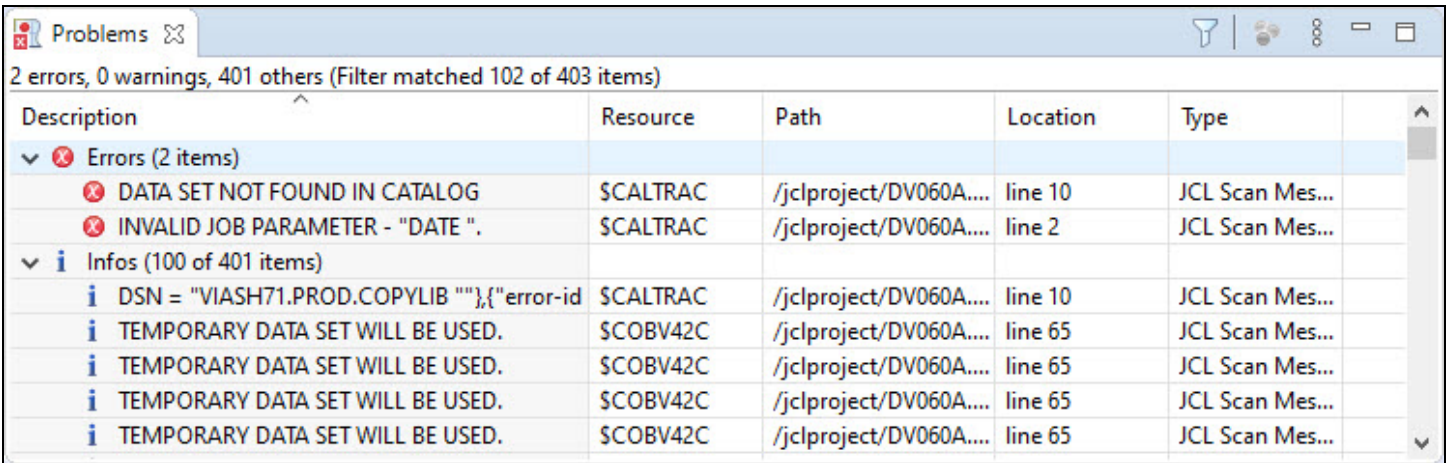
The Problems report displays the list of messages that are identified by the ASG-JOB/SCAN Plug-in.

At the top, the report summarizes the number of errors, warnings, and advisories encountered. Subsequent section displays the list of messages in each category.





Expand a category by clicking the > icon to view the list.



Information about the messages are listed in a table format containing the following columns.

Column	Description
--------	-------------

- Description** Message describing the problem
- Resource** Member name in PDS or name of sequential dataset
- Path** Partitioned dataset name or sequential dataset
- Location** Indicates the JCL line number where the error appears
- Type** *JCL Scan Message*

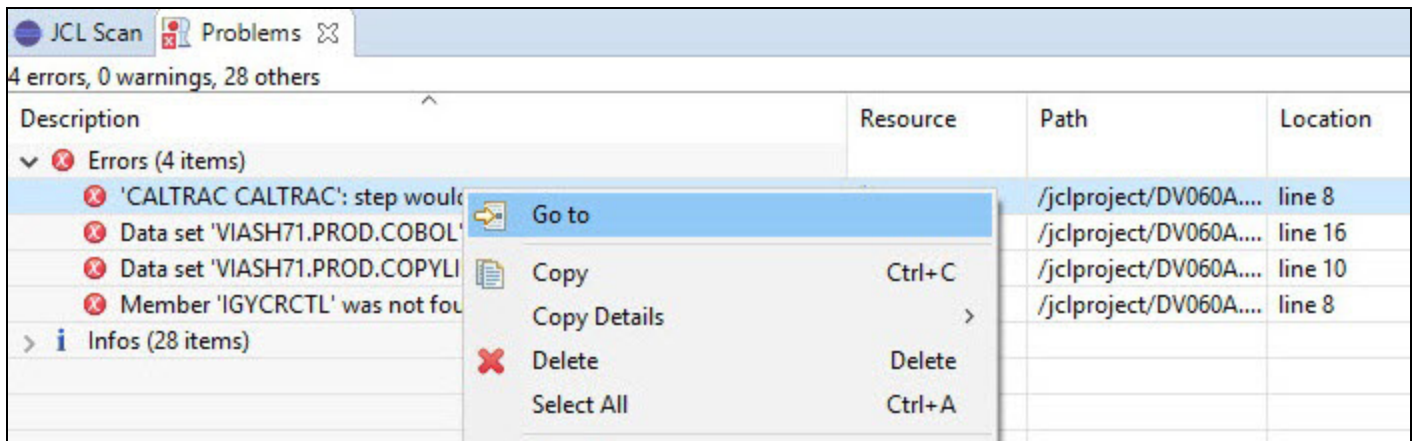
JOB/SCAN's JJFE edit macro locates a specific error in the Structured JCL Listing (SJL) output, similarly the Problems view offers a simple and quick way to locate the problem line that is associated with the message.

You can locate the problem either by:

- double-clicking on the message

Or

- right-click on the message to bring up the problems view's contextual menu and select the **Go to** option.



# Markers

Markers are small indicators that appear on the right and left margins of the JCL editor after a file is validated. These indicators enable you to quickly identify problems in your file. With these markers, user can navigate between the problem lines and identify the highest severity in a file.

Markers are categorized as follows:

- Errors
- Warnings
- Advisories (Information)



By default, only the error markers are displayed in the right and left margins. You can choose to show / hide markers as per your preference. See [Show or Hide Markers](#) for more information.

## Left-margin Markers

Markers on the left-margin are placed next to the lines that are associated with some problems. These markers move with the source lines when the content in the JCL editor is scrolled.

The left-margin can have different icons with the following meaning:

Icon	Description
	Errors
	Warnings
	Advisories (Information)

```

XMIT2AKR
//XMIT2AKR JOB (DEVCTY,282300,SRT,00), 'DUMP XMIT',CLASS=A,MSGCLASS=X, 00010000
//          NOTIFY=&SYSUID,TIME=(4,30),PRTY=6,REGION=4096K          00020000
//*                                                    00030000
//DUMP      EXEC PGM=DUMPXMIT,PARM='LOGICAL,TERSE'                00040000
//STEPLIB  DD DSN=DV060A.DEVL.LOAD,DISP=SHR                        00050000
//SYSIN    DD DSN=CUSTDUMP.T768917.XMI,DISP=SHR                    00060000
//SYSPRINT DD SYSOUT=*                                           00070000
//SYSOUT   DD DSN=DV060A.HUGE.AKR,DISP=SHR                          00080000
//*YSOUT   DD DSN=DV060A.TEST.HEXDUMP.DUMPXMIT,DISP=(NEW,CATLG),  00090000
//*                               UNIT=SRTDA,SPACE=(CYL,(1,1))      00100000
//*

```

When you hover the mouse cursor over a marker, a pop-up shows the description of the problem(s) associated with the line.

```

XMIT2AKR
//XMIT2AKR JOB (DEVCTY,282300,SRT,00), 'DUMP XMIT',CLASS=A,MSGCLASS=X, 00010000
//      NOTIFY=&SYSUID,TIME=(4,30),PRTY=6,REGION=4096K 00020000
//* 00030000
//DUMP EXEC PGM=DUMPXMIT,PARM=' LOGICAL,TERSE' 00040000
//STEPLIB DD DSN=DV060A.DEVL.LOAD,DISP=SHR 00050000
Data set 'CUSTDUMP.T768917.XMI' was not found in catalog. 00060000
//SYSPRINT DD SYSOUT=* 00070000
//SYSOUT DD DSN=DV060A.HUGE.AKR,DISP=SHR 00080000
//*YSOUT DD DSN=DV060A.TEST.HEXDUMP.DUMPXMIT,DISP=(NEW,CATLG), 00090000
//* UNIT=SRTDA,SPACE=(CYL,(1,1)) 00100000
//*

```

## Right-margin Markers

Markers on the right-margin show the relative locations of the tagged lines in the file. As a result, the markers are still visible when the margin shrinks to fit the screen. These markers do not move when the content in the editor is scrolled. Instead, clicking any marker in the right-margin would highlight the line associated with that marker. This allows you to quickly step through the problem lines in the editor by simply clicking the markers one by one.

Markers on the right-margin are color coded according to the message type:

Color	Description
Blue	Advisory messages
Yellow	Warning messages
Red	Error messages

```

XMIT2AKR
//XMIT2AKR JOB (DEVCTY,282300,SRT,00), 'DUMP XMIT',CLASS=A,MSGCLASS=X, 00010000
//      NOTIFY=&SYSUID,TIME=(4,30),PRTY=6,REGION=4096K 00020000
//* 00030000
//DUMP EXEC PGM=DUMPXMIT,PARM=' LOGICAL,TERSE' 00040000
//STEPLIB DD DSN=DV060A.DEVL.LOAD,DISP=SHR 00050000
Data set 'CUSTDUMP.T768917.XMI' was not found in catalog. 00060000
//SYSPRINT DD SYSOUT=* 00070000
//SYSOUT DD DSN=DV060A.HUGE.AKR,DISP=SHR 00080000
//*YSOUT DD DSN=DV060A.TEST.HEXDUMP.DUMPXMIT,DISP=(NEW,CATLG), 00090000
//* UNIT=SRTDA,SPACE=(CYL,(1,1)) 00100000
//*

```

When you hover the mouse cursor over a marker, a pop-up shows the description of the problem(s) associated with the line.

```

XMIT2AKR
//XMIT2AKR JOB (DEVCTY,282300,SRT,00),'DUMP XMIT',CLASS=A,MSGCLASS=X, 00010000
//      NOTIFY=&SYSUID,TIME=(4,30),PRTY=6,REGION=4096K 00020000
//* 00030000
//DUMP EXEC PGM=DUMPXMIT,PARM='LOGICAL,TERSE' 00040000
//STEPLIB DD DSN=DV060A.DEVL.LOAD,DISP=SHR 00050000
//SYSIN DD DSN=CUSTDUMP.T768917.XMI,DISP=SHR 00060000
//SYSPRINT DD SYSOUT=* 00070000
//SYSOUT DD DSN=DV060A.HUGE.AKR,DISP=SHR 00080000
//*YSOUT DD DSN=DV060A.TEST.HEXDUMP.DUMPXMIT,DISP=(NEW,CATLG), 00090000
//* UNIT=SRTDA,SPACE=(CYL,(1,1)) 00100000
//*

```

A tooltip is displayed over the SYSOUT DD statement: "Data set 'DV060A.HUGE.AKR' has been migrated."

The square that appears at the top right-margin shows the highest severity found in the file. Messages have the following severity levels:

Severity	Color	Description
High	Red	Error messages
Medium	Yellow	Warning messages
Low	Blue	Advisory messages

The square is not displayed if there are no messages associated with the file.

When you hover the mouse cursor over the square, a pop-up shows the total number of severities encountered in the current file.

```

XMIT2AKR
//XMIT2AKR JOB (DEVCTY,282300,SRT,00),'DUMP XMIT',CLASS=A,MSGCLASS=X, 00010000
//      NOTIFY=&SYSUID,TIME=(4,30),PRTY=6,REGION=4096K 00020000
//* 00030000
//DUMP EXEC PGM=DUMPXMIT,PARM='LOGICAL,TERSE' 00040000
//STEPLIB DD DSN=DV060A.DEVL.LOAD,DISP=SHR 00050000
//SYSIN DD DSN=CUSTDUMP.T768917.XMI,DISP=SHR 00060000
//SYSPRINT DD SYSOUT=* 00070000
//SYSOUT DD DSN=DV060A.HUGE.AKR,DISP=SHR 00080000
//*YSOUT DD DSN=DV060A.TEST.HEXDUMP.DUMPXMIT,DISP=(NEW,CATLG), 00090000
//* UNIT=SRTDA,SPACE=(CYL,(1,1)) 00100000
//*

```

A tooltip is displayed over the top right margin: "Errors: 1 Warnings: 1"

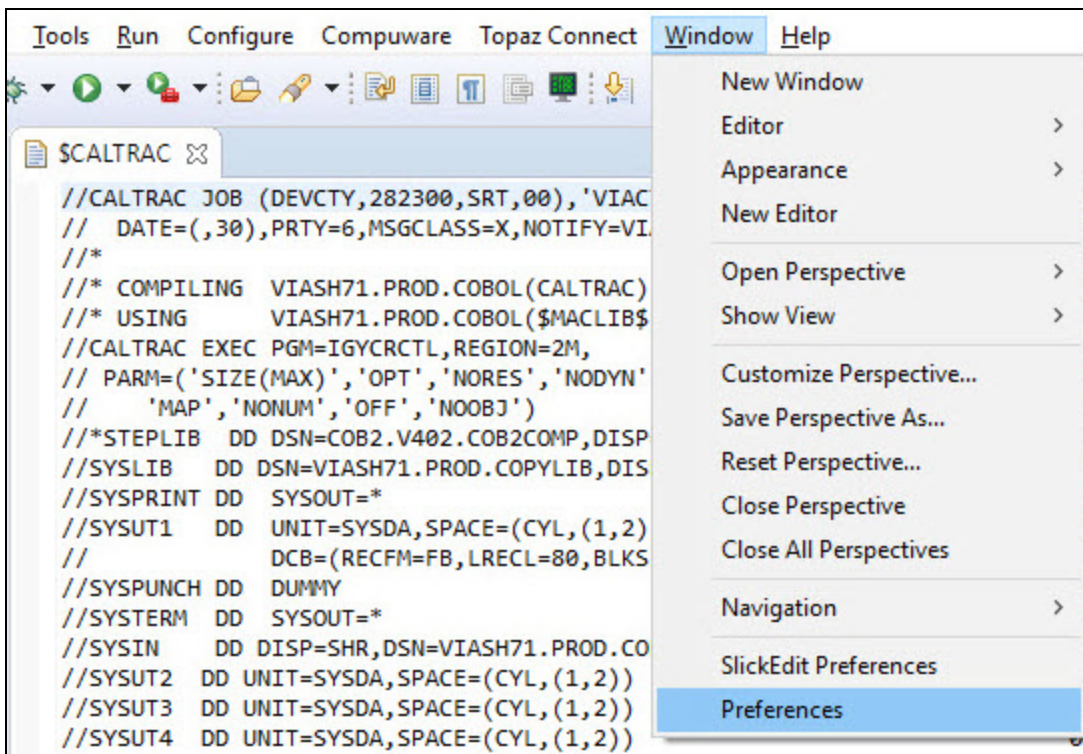
## Show or Hide Markers

Markers like errors, warnings and infos are used to quickly identify the problems in a file. You might find these markers useful when you are validating a JCL file. You may choose how you want these markers to be displayed through the

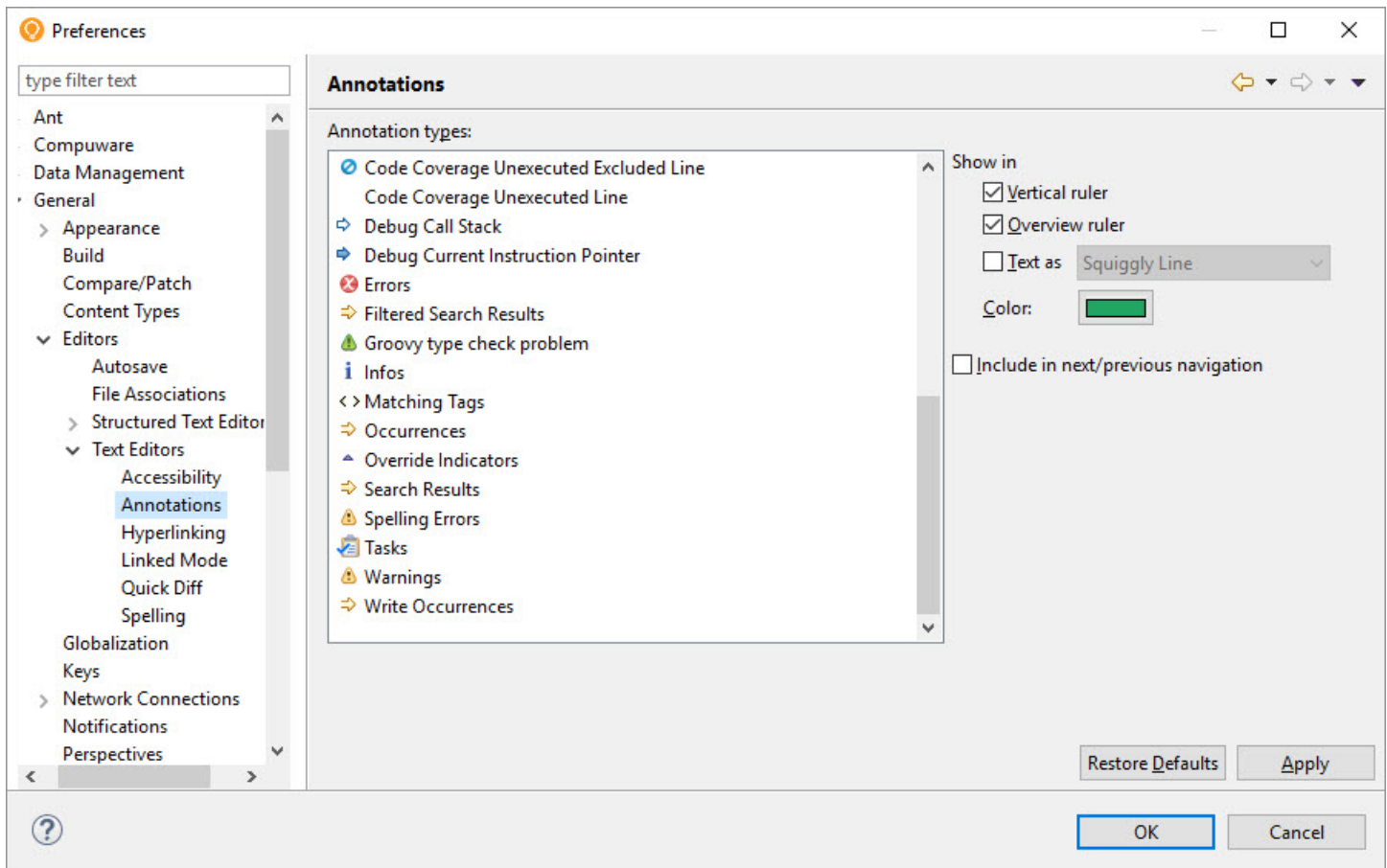
Preferences dialog.

**To show or hide markers:**

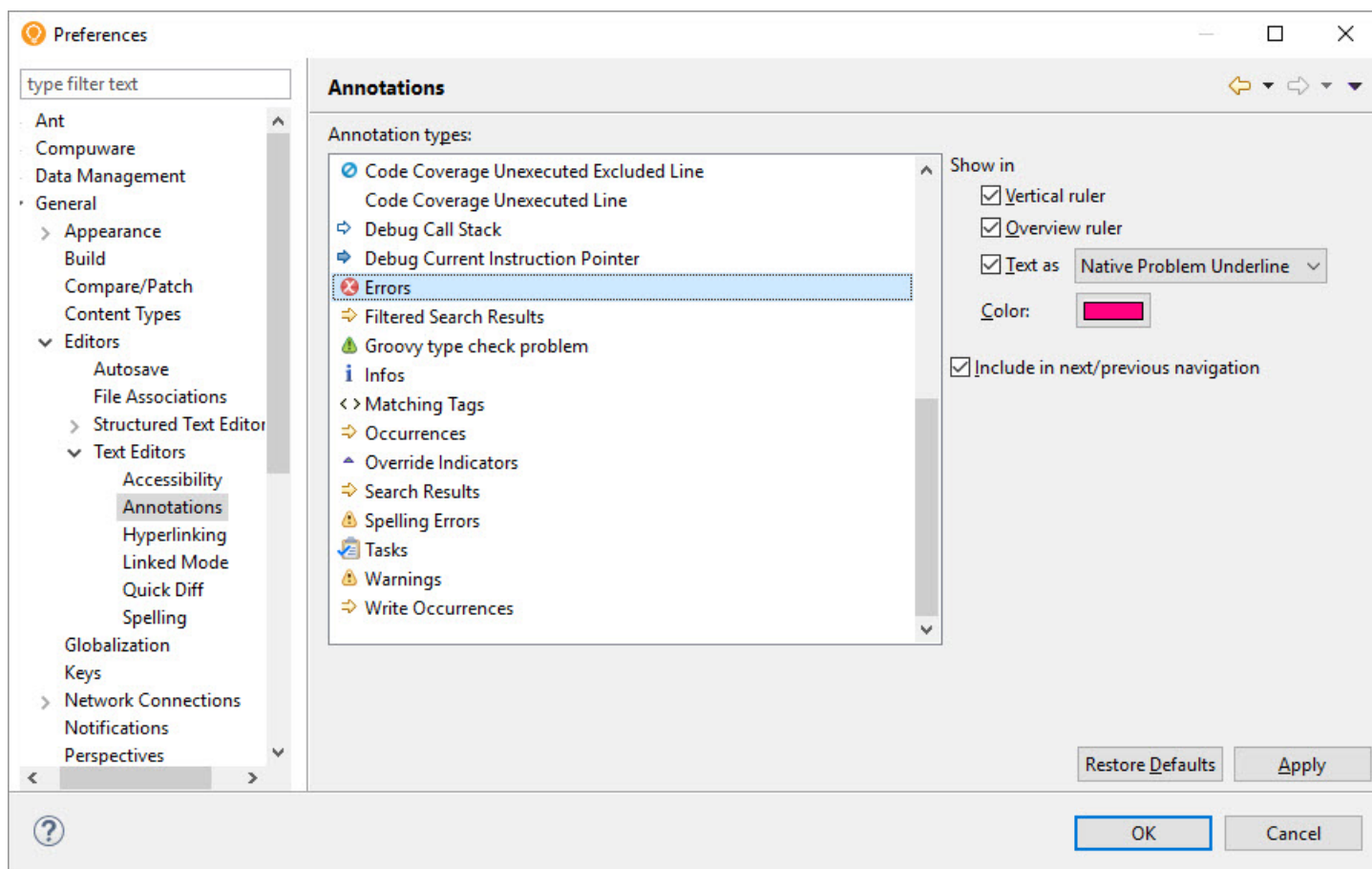
1. From the **Window** menu, select **Preferences**.



2. In the **Preferences** dialog, select **General > Editors > Text Editors > Annotations**.



3. In the **Annotations** page, under the **Annotations types**, select **Errors / Infos / Warnings** as per your requirement.



4. Under **Show in**, select or deselect the following options:

Options	Description
Vertical ruler	This option control the markers on the left-margin.
Overview ruler	This option control the markers on the right-margin.
Text as	<p>This option control the format of the tagged lines in the JCL editor region. You can select on of the following formats:</p> <ul style="list-style-type: none"> <li>• Box</li> <li>• Dashed Box</li> <li>• Highlighted</li> <li>• Native Problem Underline</li> <li>• Squiggly Line</li> <li>• Underlined</li> <li>• Vertical Bar</li> </ul>



5. After choosing your preference, click **Apply** to save the changes or click **Apply and Close** to save the changes and close the **Annotations** dialog.

## Keyboard Shortcuts

In this topic you can find the list of all the keyboard shortcuts for different tasks performed in ASG-JOB/SCAN Plug-in. Also, the topic explains the procedure to customize the default shortcuts.

The following table shows the shortcut keys used in ASG-JOB/SCAN Plug-in:

Task	Default Shortcut
Scan JCL	Ctrl + 8
Scan and submit JCL	Ctrl + 9
Save to z/OS	Ctrl + 5
Save to z/OS As...	Ctrl + Shift + 5

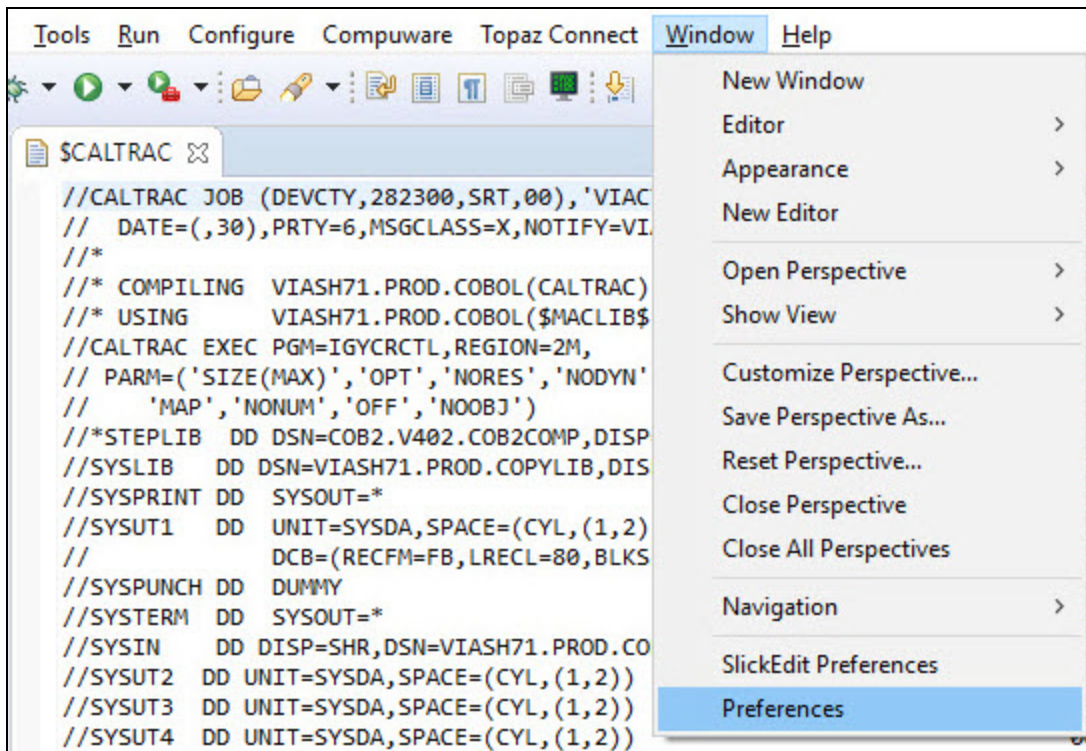
---

## Customizing Keyboard Shortcuts

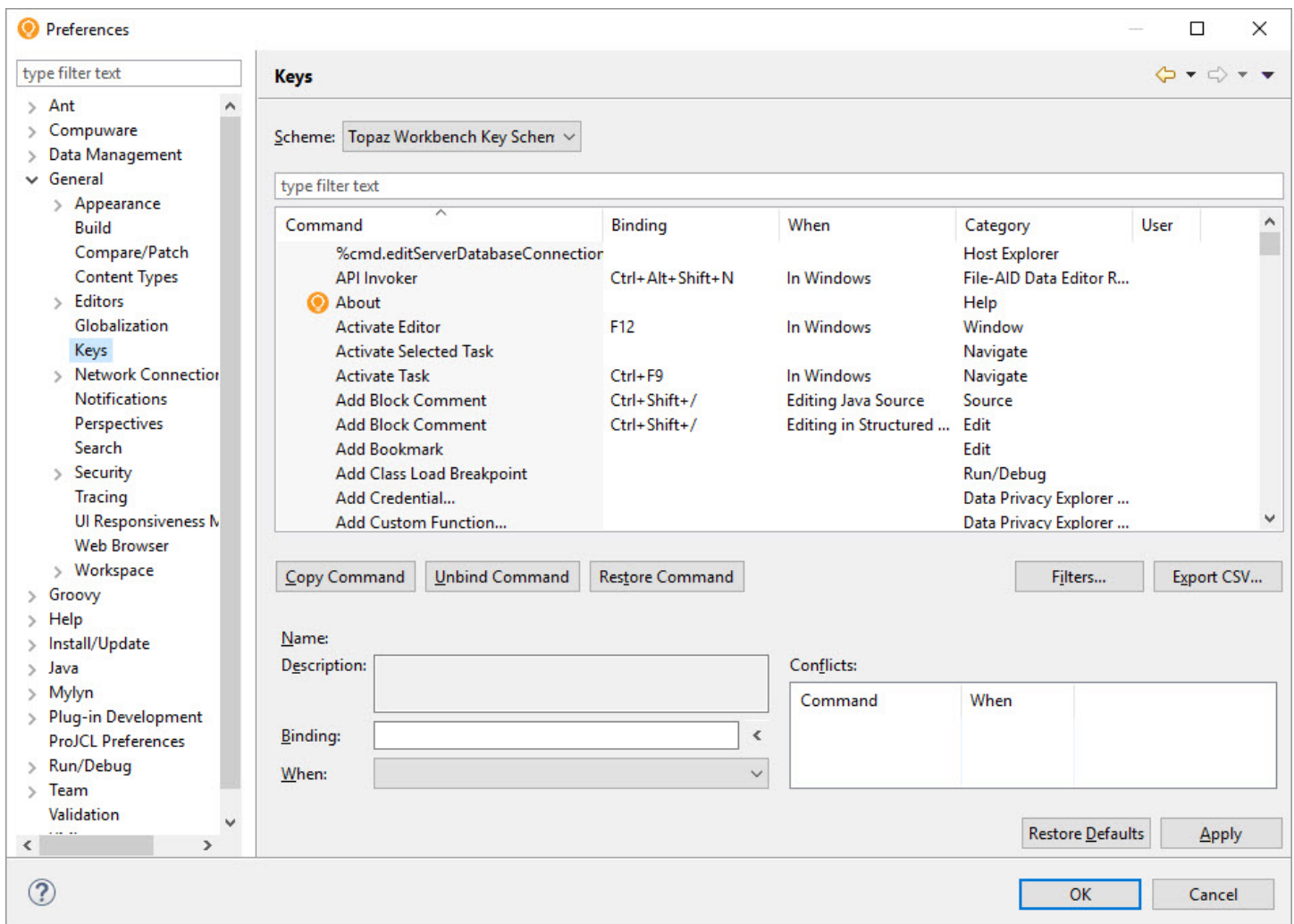
The key combinations for the existing shortcuts can be disabled or modified as required.

**To customize shortcuts:**

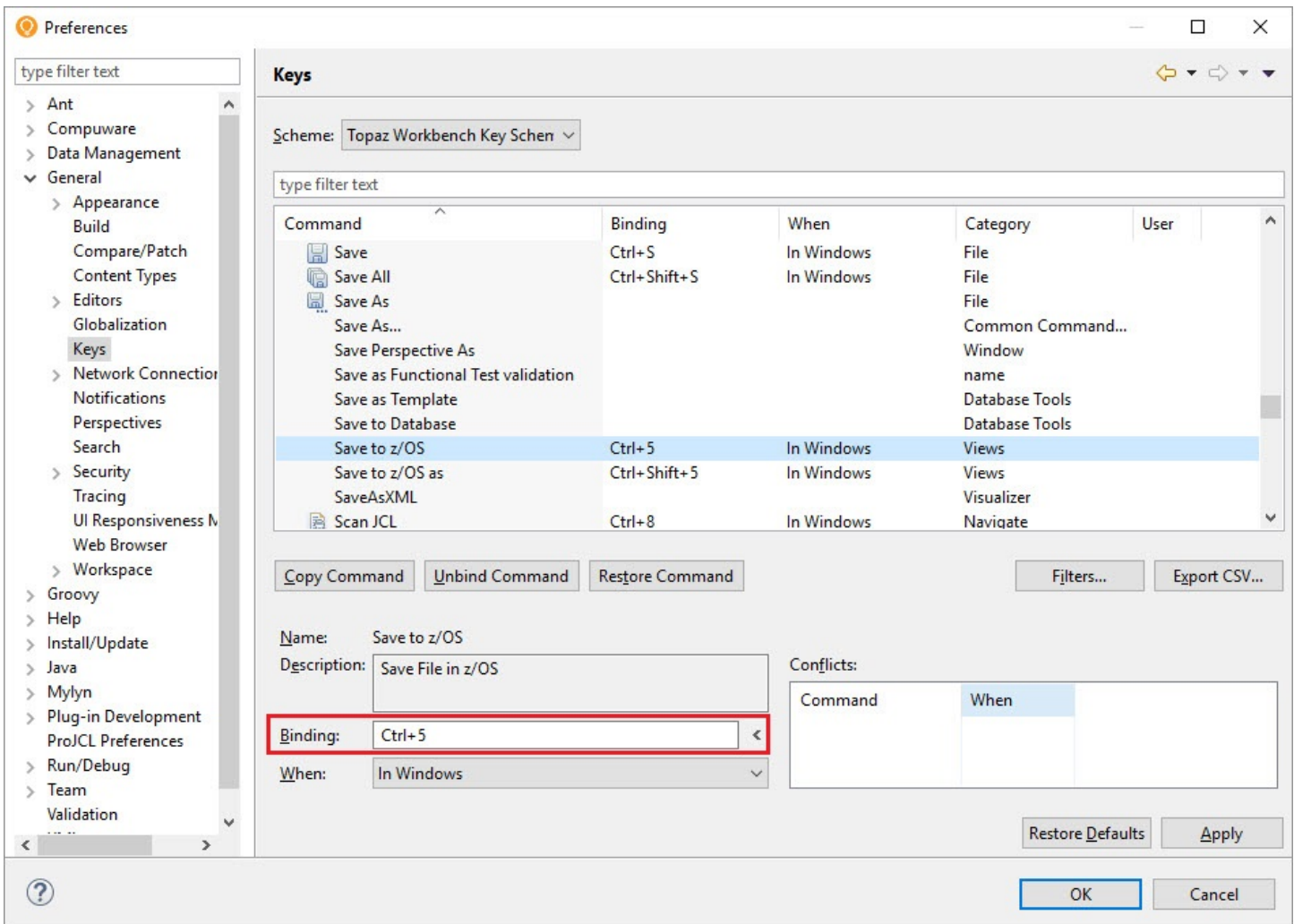
1. From the **Window** menu, select **Preferences**.



2. In the **Preferences** dialog, select **General > Keys**.



- In the **Keys** page, select the **Command** that you like to change. The values associated with the command is populated in the fields.



4. To change the key combinations, edit the **Binding** field.
5. If required, you can edit the values in the other fields.
6. click **Apply** to save the changes or click **Apply and Close** to save the changes and close the **Preferences** dialog.

# ASG-JOB/SCAN REST API Documentation

The ASG-JOB/SCAN REST API service is used to interact with ASG-JOB/SCAN REST server. It gives the ability to scan the JCLs from any application. The ASG-JOB/SCAN REST API service is defined using the OpenAPI 3.0 specification.

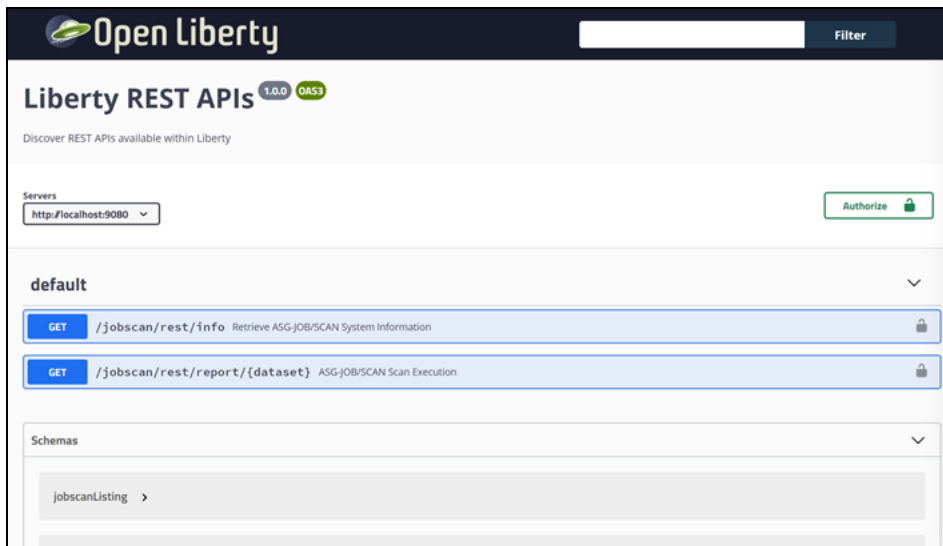
The ASG-JOB/SCAN REST API user interface is accessible at the URL: `http://<Host_Name>:<Port>/api/explorer/`

wherein:

*Host\_Name* indicates the host name or IP address of the z/OS system

*Port* indicates the port number of the host machine on which the JOB/SCAN REST API service is running

Example: `http://mvssyse.asg.com:60080/api/explorer/`

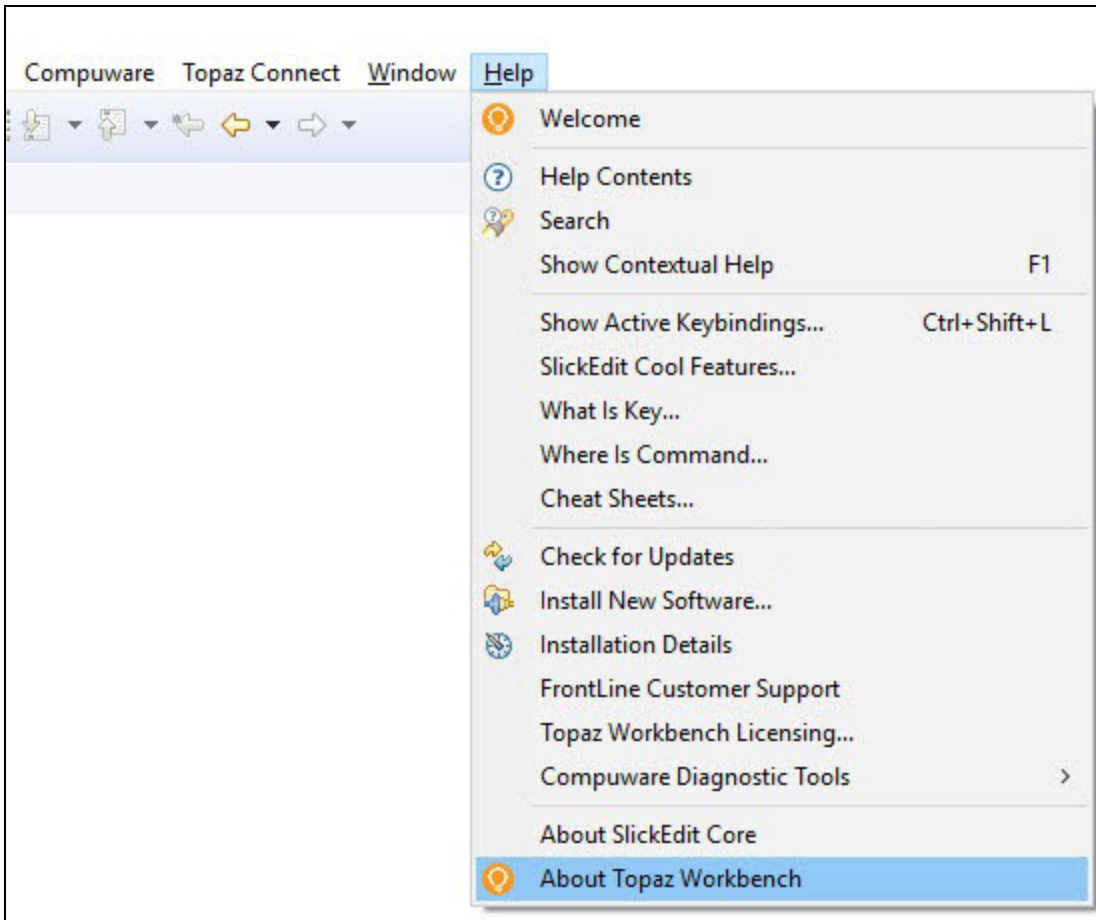


# Uninstallation

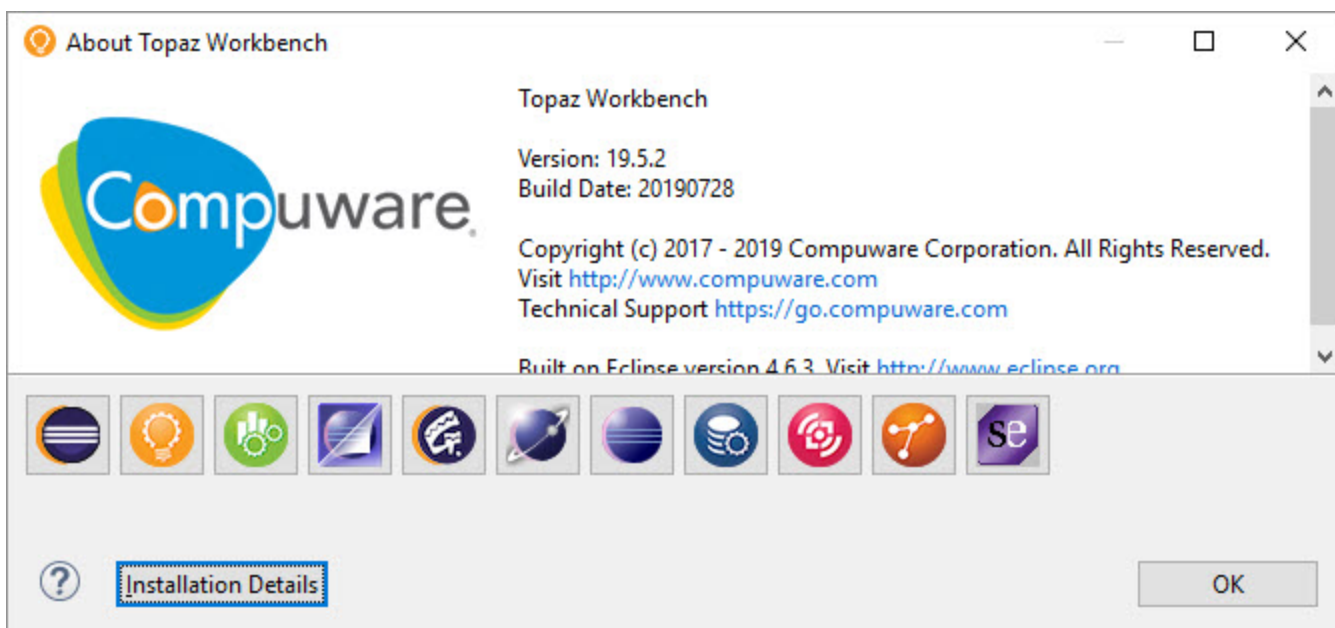
Before installing the new version of ASG-JOB/SCAN Plug-in, it is recommended to remove any prior versions from the server.

## To uninstall ASG-JOB/SCAN Plug-in:

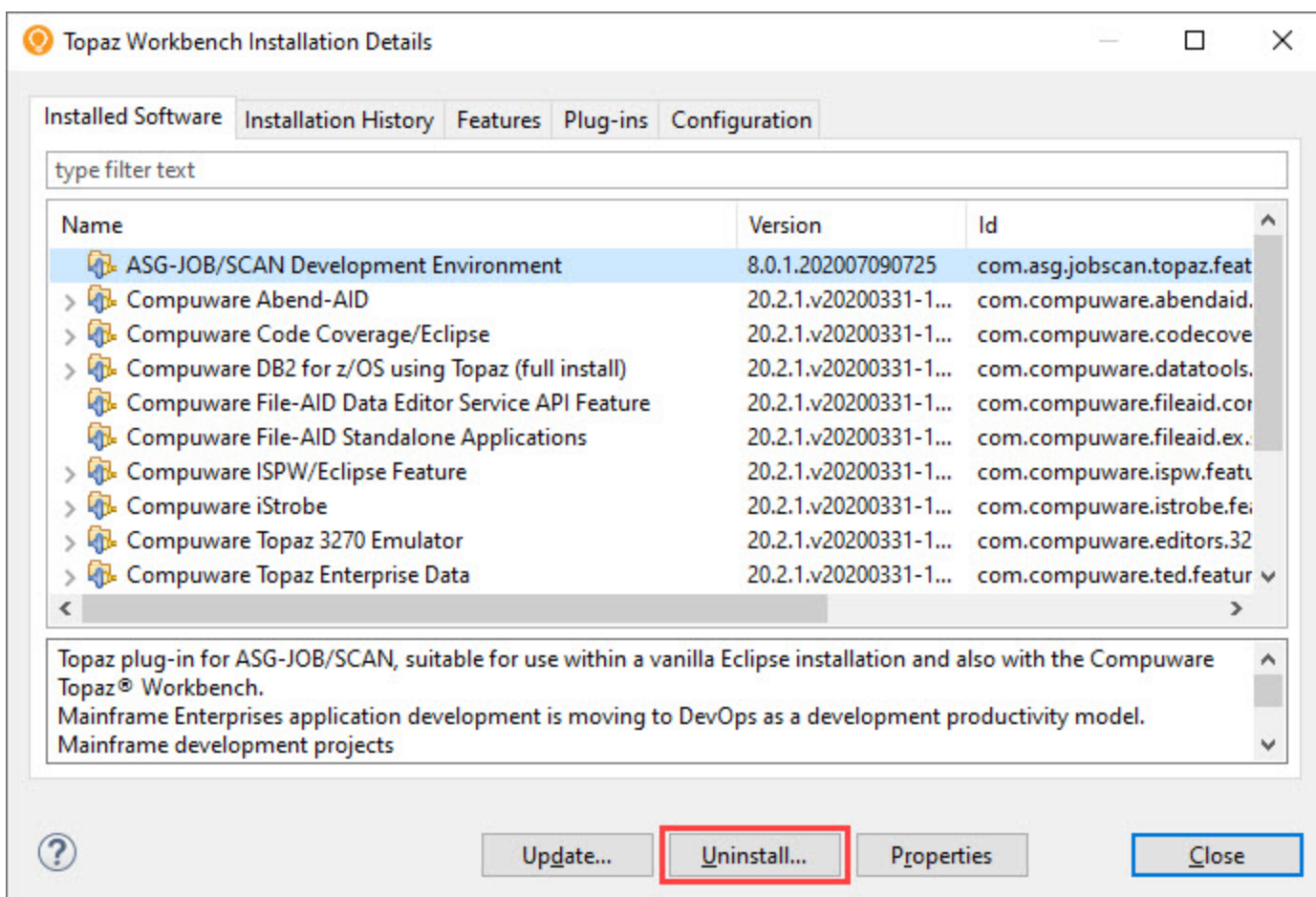
1. Start Topaz Workbench.
2. From the menu, select **Help > About Topaz Workbench**.



3. In the **About Topaz Workbench** dialog, click **Installation Details**.

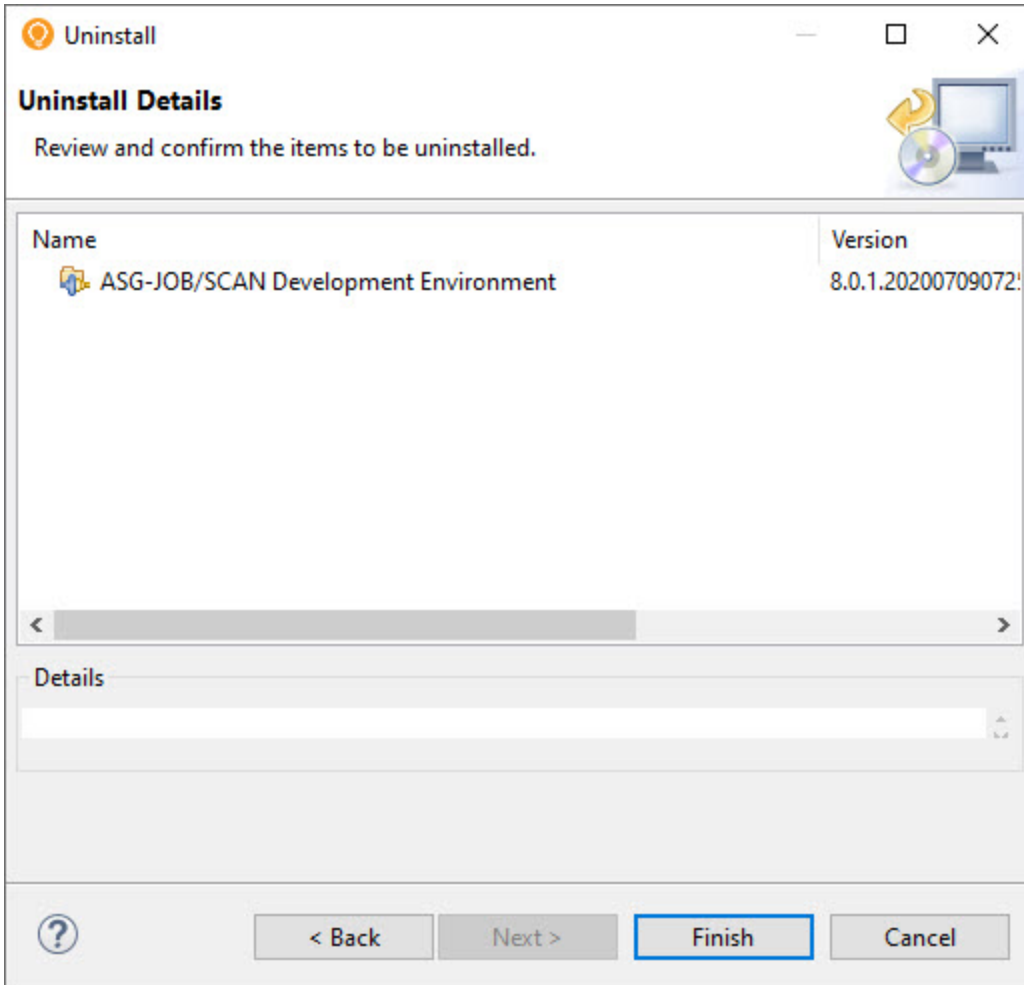


4. In the **Installed Software** tab, select **ASG-JOB/SCAN Development Environment** and click **Uninstall...**

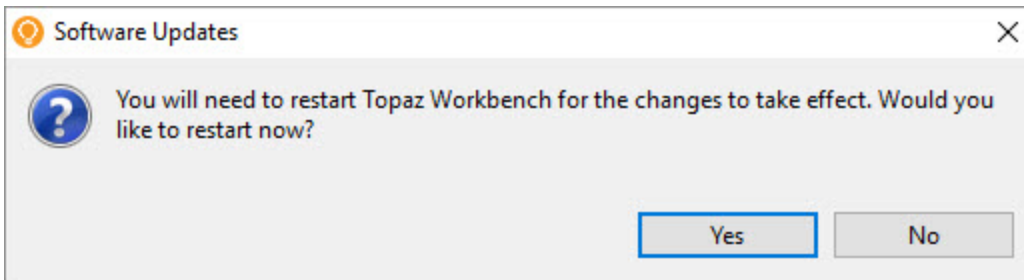




5. Review the uninstall items and click **Finish**.



6. When the uninstallation is complete, you are prompted to restart the Topaz application. Click **Restart Now**.



**To verify if the plug-in has been uninstalled:**

1. Start Topaz Workbench.
2. From the menu, select **Help > About Topaz Workbench**.
3. In the **About Topaz Workbench** dialog, click **Installation Details**.

4. In the **Installed Software** tab, you should not see **ASG-JOB/SCAN Development Environment** in the list of installed software. This indicates that ASG-JOB/SCAN Plug-in has been uninstalled.