CA JCLCheck™ Workload Automation

Tutorial

Version 12.0.00

CA Technologies Product References

This document references the following CA Technologies products:

- CA Workload Automation CA 7® Edition (CA WA CA 7 Edition)
- CA Workload Automation Restart Option for z/OS Schedulers (CA WA Restart Option for z/OS Schedulers)
- CA Scheduler® Job Management (CA Scheduler)
- CA Workload Automation ESP Edition (CA WA ESP Edition)
- CA APCDOC™ Automated Job Documentation (CA APCDOC)
- CA Dispatch™ (CA Dispatch)
- CA ACF2™ (CA ACF2)
- CA Top Secret® (CA Top Secret)
- CA ASM2® Backup and Restore (CA ASM2)
- CA 1® Tape Management (CA 1)
- CA TLMS® Tape Management (CA TLMS)
- CA Roscoe® Interactive Environment (CA Roscoe)
- CA Librarian® (CA Librarian)
- CA Panvalet® (CA Panvalet)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

Resolve External Variables

Updated the CA JCLCheck main menu with a new RESOLVE option for the following topics:

- Setting up CA JCLCheck options (see page 18)
- Viewing options through CA JCLCheck invocation panels (see page 22)
- Scanning a JCL member (see page 23)
- Report 1 - Listing of Jobstream JCL (see page 28)
- Check a User ID for Security (see page 57)
- Set Up CA APCDOC Integration (see page 70)
- Set Up CA Scheduler Integration (see page 78)
- Set Up CA 7 Integration (see page 80)

The following topics are all new for the resolve external variables enhancement:

- About External Variables (see page 81)
- How to Configure CA JCLCheck to Resolve External Variables (see page 83)
- Specify the Product and Parameters using EDCHEK ISPF (see page 84)
- Edit and Validate the Parameter Data Set using EDCHEK ISPF (see page 85)
- Specify the Product using ISPF (see page 86)
- Specify and Submit the Parameters using ISPF (see page 86)
- Specify the Product and Parameters in Batch (see page 91)
- CA WA CA 7 Edition Parameters (see page 95)
- CA WA CA 7 Edition Driver Procedures (see page 98)
- Sample Parameter Data Set for CA7 (see page 100)
- CA WA ESP Edition Parameters (see page 100)
- Sample Parameter Data Set for ESP (see page 104)
- Control-M Parameters (see page 104)

- Sample Parameter Data Set for CNTL-M (see page 107)
- Tivoli Workload Scheduler Parameters (see page 107)
- Sample Parameter Data Set for TWS (see page 112)

REXX Enhancements

- View the JCLNeat Options Panel (see page 63)—Updated the FORMAT options.

ISPF List of REXX Rules

- View the JCLNeat Options Panel (see page 63)—Updated the REXXMEM option.

Alternate User ID

- Alternate User ID (see page 60)—New.
- Setup (see page 60)—New.
- Installation Considerations (see page 62)—New.

Documentation Fixes

- Setting up CA JCLCheck Options (see page 18)—Updated the Process Control Options screen to include CSI.

Contents

Chapter 1: Introduction 11

Overview 11

Chapter 2: How to Use CA JCLCheck 13

Running CA JCLCheck in Batch 13
- Accessing member AZ1JCHK 14
- Modifying member AZ1JCHK 15
- Executing member AZ1JCHK 16

Invoking CA JCLCheck with the ISPF Dialog 16
- Accessing the CA JCLCheck SPF Menu (JCK0200) 17
- Setting up CA JCLCheck Options 18
- Setting up your job card 20
- Viewing options through CA JCLCheck invocation panels 22
- Scanning a JCL member 23
- Scanning a group of JCL members 24
- Specifying CA JCLCheck options through the invocation panels 25
- Submitting JCL to run CA JCLCheck in batch 26

Selecting Reports 27
- Report 1 - Listing of Jobstream JCL 28
- Report 2 - Listing of Merged JCL 31
- Report 3 - Data Set Cross-Reference 32
- Report 4 - Program Cross-Reference 32
- Report 5 - Report Listing 33
- Report 6 - Error Messages 33
- Report 7 - Summary Data Set Cross-Reference 34
- Report 8 - Summary Program Cross-Reference 35
- Report 9 – Summary Report Listing 36
- Report 10 - Flow Diagram 36
- Report 11 - Procedure Cross-Reference 40
- Audit Report 41

Defining Syntax Checking and Message Severity 42
- Viewing Error Messages 43
- Eliminating Warning Messages 44

Defining Runtime 45
- Simulate Runtime Errors 46
- Perform Runtime Checking 46

Deactivate Runtime Checking 47
Invoking CA JCLCheck with EDCHEK 49
Invoke EDCHEK as an Edit Macro 50
Change Runtime Options Prior to EDCHEK Invocation 51
Specify Options 52
View Reports 53
Continue Editing JCL 54
Invoking CA JCLCheck with SUBCHEK 55
Setting-Up Security 56
Check a User ID for Security 57
Create an Error Situation to Test Security 59
Alternate User ID 60
Using the JCLNeat Component to Reformat JCL 62
View the JCLNeat Options Panel 63
Re-invoke the JCLNeat Options Panel 64
View the Results 64
Using CA Roscoe 65
Invoke CA JCLCheck from your Active Work Space (AWS) 66
Invoke CA JCLCheck as a Roscoe Program Facility (RPF) 66
Invoke JCLNeat under CA Roscoe 69
Integrating Products 70
Set Up CA APCDOC Integration 70
Set Up CA ASM2 Integration 76
Set Up CA Dispatch Integration 77
Set Up CA TLMS Integration 77
Set Up CA Easyproclib Integration 77
Set Up CA Scheduler Integration 78
Set Up CA 1 Integration 79
Set Up DB2 Integration 79
Set Up IMS Integration 79
Set Up CA WA CA 7 Edition Integration 80
Set Up CA WA Restart Option for z/OS Schedulers Integration 81
About External Variables 81
Executing JCL DD Statements 112

Chapter 3: How to Use Job Control Standards 117

Masks 118
Adding a Standard 118
Creating the Standard Name 119
Selecting the Statement and Parameter Type 119
Defining the Standards Rule 120

Selecting the Action 121
Defining the Action 122
Confirming the Standard 123
Compiling the Standard 123
Using the Standard 124
Printing the Standard 125
Adding a Standard with More Than One Rule 126
Creating the Standard Name 127
Selecting the Statement and Parameter Type 127
Defining the Standards Rule 128
Selecting the Action 129
Defining the Action 130
Confirming the Standard 131
Selecting a Statement and Parameter/Subparameter Type 131
Defining a Standards Rule 132
Selecting the Action 133
Defining the Standards Rule 133
Defining the Action 134
Confirming the Standard 135
Compiling the Standard 135
Using the Standard 135
Printing the Standard 136
Deleting a Standard 136
Deleting a Rule Related to an Error Message 137
Deleting Action Rules from a Standard 140

Chapter 4: CA JCLCheck REXX Programming Interface 145

REXX EXECs 145
Accessing the CAZ1REXX Template 145
Creating a REXX EXEC 145
Submitting JCL Using the REXX EXEC 146
The INITIAL PROCESSING and DO WHILE Loop 146
The $CAJCL_ERROR Subroutine 147
REXX EXEC #1 147
Modifying the INITIAL PROCESSING Subroutine 148
Modifying the JOB Processing Subroutine 148
Modifying the EXEC Processing Subroutine 149
Modifying the DD Processing Subroutine 150
Modifying the END-OF-STEP Processing Subroutine 150
Modifying the END-OF-JOB Processing Subroutine 151
Examining Report 2 - Listing of Merged JCL 152

Examining Report 6 - Error Messages 153
Sample REXX Output Using EDCHEK 155
REXX EXEC #2 156
Modifying the INITIAL PROCESSING Subroutine 156
Modifying the JOB Processing Subroutine 157
Modifying EXEC Processing Subroutine 158
Modifying the DD Processing Subroutine 160
Modifying the END-OF-STEP Processing Subroutine 161
Modifying the END-OF-JOB Processing Subroutine 162
Examining Report 2 - Listing of Merged JCL 163
Examining Report 6 - Error Messages 164

Chapter 5: JCLNeat REXX Programming Interface 167

REXX EXECs 167
Accessing the CAZ1NREX Template 167
Creating a REXX EXEC 168
Submitting JCL Using the REXX EXEC 168
The INITIAL PROCESSING and DO WHILE Loop 168
The $CAJCL_ERROR Subroutine 170
REXX EXEC 170
Modifying the INITIAL PROCESSING Subroutine 171
Modifying the JOB Processing Subroutine 171
Modifying the EXEC Processing Subroutine 172
Modifying the DD Processing Subroutine 172
Modifying the END-OF-STEP Processing Subroutine 174
Modifying the END-OF-JOB Processing Subroutine 175
RAW Data Processing Subroutine 176
JCLNeat Original JCL Report 177
Reformatted JCL Report 178

Chapter 1: Introduction

Overview

This tutorial is for CA JCLCheck administrators and other users of the product. It provides step-by-step examples of how to:

- Use the fundamental features of CA JCLCheck
- Use the Job Control Standards panels
- Use the CA JCLCheck REXX programming interface
- Use the JCLNeat REXX programming interface

This tutorial explains how to invoke and use specific features of CA JCLCheck. The features are explained in a concise format to help you learn how to use CA JCLCheck.

Important! Before using this Tutorial, review chapters 1 and 2 of the Command Reference Guide to use your own JCL as input to CA JCLCheck.

Chapter 2: How to Use CA JCLCheck

This chapter provides you with a step-by-step approach to using the fundamental features of CA JCLCheck. It shows you how to invoke CA JCLCheck in foreground and batch, use the EDCHEK and SUBCHEK features, invoke CA JCLCheck in different environments, set up your default options, select JCL members, define syntax checking, and produce CA JCLCheck reports.

Each feature contains an overview, a list of the feature's functions, instructions to use the feature, and a paragraph that tells you where you can find additional information about the feature.

The JCL DD statements that are available to run CA JCLCheck in batch are described at the end of this chapter. For information, see the topic Executing JCL DD Statements (see page 112).

Important! Using some of the features in this chapter requires the installation of those features. The feature sections in the chapter refer you to the necessary steps to view in the Installation Guide. It is assumed that CA JCLCheck is already installed at your site, according to your site's needs. This may or may not allow you to use certain features.

This section contains the following topics:

Running CA JCLCheck in Batch (see page 13)
Invoking CA JCLCheck with the ISPF Dialog (see page 16)
Selecting Reports (see page 27)
Defining Syntax Checking and Message Severity (see page 42)
Defining Runtime (see page 45)
Invoking CA JCLCheck with EDCHEK (see page 49)
Invoking CA JCLCheck with SUBCHEK (see page 55)
Setting-Up Security (see page 56)
Using the JCLNeat Component to Reformat JCL (see page 62)
Using CA Roscoe (see page 65)
Integrating Products (see page 70)
Executing JCL DD Statements (see page 112)

Running CA JCLCheck in Batch

CA JCLCheck execution in the background (batch) permits you to display CA JCLCheck output on the JES spool and also use the JES subsystem to print output on local and channel attached printers.

To invoke CA JCLCheck in batch mode, use the sample JCL provided in member AZ1JCHK in the CAZ2JCL data set. This member uses the CAZ1JCHK JCL procedure located in CAZ2PROC to execute the JCLCheck program.

Once you tailor this JCL, all you need to do is submit it for processing. When processing is complete, the primary reports appear in the SYSPRINT DD as referenced in the CAZ1JCHK procedure library member.

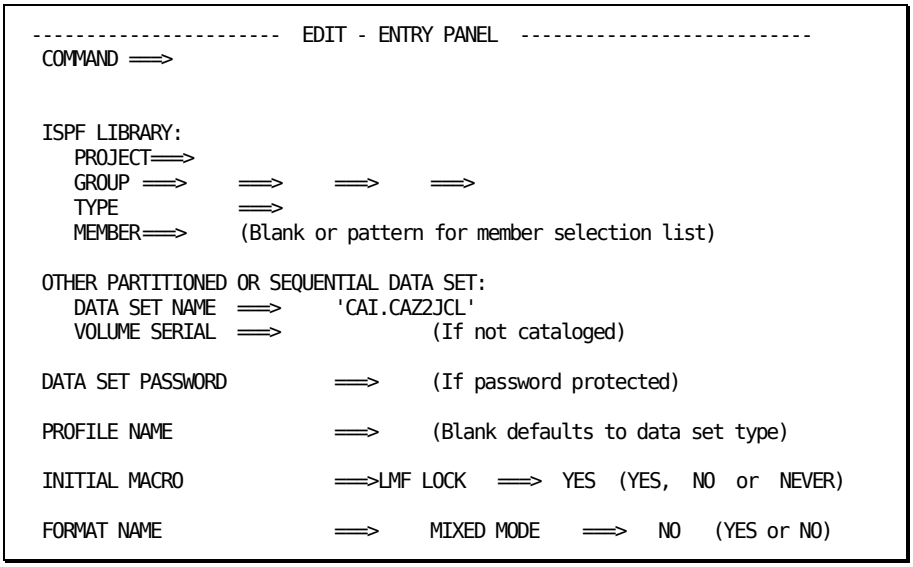
Note: The CAZ1JCHK procedure is usually prepared by the person responsible for installing CA JCLCheck. This JCL usually remains constant. The programmer using CA JCLCheck usually tailors the PROC symbolic in the AZ1JCHK member.

To tailor the AZ1JCHK JCL member, complete the following three steps:

- Accessing member AZ1JCHK (see page 14)
- Modifying member AZ1JCHK (see page 15)
- Executing member AZ1JCHK (see page 16) to scan a group of JCL members

Accessing member AZ1JCHK

1. Invoke ISPF edit and enter the data set name that contains this member.
2. Select member AZ1JCHK, and press ENTER.



```
EDIT --------------- CAI.APCMTL.CAZ2JCL----------- ROW   00145  OF  00510
COMMAND ===>                                                  SCROLL ===> PAGE

NAME       VV.MM    CREATED     CHANGED    SIZE    INIT   MOD    ID
UZ26203
Z1EXGCHK
 S     AZ1JCHK
Z1EXNEAT
Z1XSMFA
Z1JCSCMP
Z1JCSRPT
Z262DFLT
```

Note: This data set contains the AZ1JCHK JCL member for an installed CA JCLCheck system.

Modifying member AZ1JCHK

1. Enter your job statement information in the JOB statement.
2. Enter processing options beneath the OPTS DD statement.
3. On the SYSIN DD statement, enter the data set name and JCL member name that you want CA JCLCheck to analyze. See comment in the AZ1JCHK member of other SYSIN input options.
4. Enter **SAVE** on the COMMAND line, and press enter to save the changes to the AZ1JCHK member.
5. Enter **SUB** on the COMMAND line, and press enter to submit the JCL member for CA JCLCheck checking.

```
EDIT ----------- CAI.CAZ2JCL(AZ1JCHK) - 01.00 -------- COLUMNS 001 072
COMMAND ===>SAVE      SUB                                SCROLL ===> PAGE

 //AZ1JCHK  JOB  '41400000',ALYSSA,CLASS=K,MSGCLASS=X
 //Z1EXJCHK  EXEC  CAZ1JCHK,
 //          OPTION='O(OPTS)'
 //OPTS  DD  *                              YOUR RUNTIME OPTIONS
 EASYPROC,  ER(S)
 /*
 //SYSIN  DD  DSN=CAI.CAZ2JCL(CAZ1SAMP),DISP=SHR
 //
```

Note: This member contains comments (not shown here) that explain how to process partitioned data sets and library files.

Executing member AZ1JCHK

1. Enter the INCLUDE option in the processing options beneath the OPTS DD statement, followed by a group of member names. Enter **PDS**, **PAN**, or **LIB**, based on the type of JCL library you are using.
2. Modify your SYSIN DD statement to point to a PDS, CA Panvalet, or CA Librarian master file.
3. Enter **SAVE** on the COMMAND line, and press enter to save the changes to the AZ1JCHK member.
4. Enter **SUB** on the COMMAND line, and press enter to submit the JCL member for CA JCLCheck checking.

```
EDIT --- CAI.APCMTL.CHQAJ661.CAZ2JCL(AZ1JCHK) - 01.00 ----- COLUMNS 001 072
COMMAND ===>SAVE      SUB                                        SCROL==> PAGE

 //AZ1JCHK  JOB  '41400000',ALYSSA,CLASS=K,MSGCLASS=X
 //Z1EXJCHK  EXEC  CAZ1JCHK,
 //          DSNQUAL='CAI',
 //          DSNLOAD='CAILOAD',
 //          OPTION='O(OPTS)'
 //OPTS   DD  *                          YOUR RUNTIME OPTIONS
 EASYPROC,  ER(S), INCLUDE(CAZ1SAMP,CAZ1SE1,CAZ1SE2,CAZ1SE3),PDS
 /*
 //SYSIN  DD  DSN=CAI.CAZ2JCL,DISP=SHR
 //
```

Note: For information about the access and modification of member AZ1JCHK, see the *Installation Guide*, or see Invoking with the Batch Facility in the *Command Reference Guide*.

Invoking CA JCLCheck with the ISPF Dialog

Invoking CA JCLCheck with the ISPF dialog illustrates the CA JCLCheck Dialog Manager interface. You can use this interface with IBM ISPF/PDF under z/OS to do the following tasks:

- Set up your CA JCLCheck options
- Run CA JCLCheck in the foreground
- Submit a job to execute CA JCLCheck in the background
- Select jobs that you want CA JCLCheck to scan

If you execute CA JCLCheck in the foreground, you can:

- Request CA JCLCheck to submit error-free jobs for execution
- Review the output using BROWSE
- Generate hardcopy output using the PRINTOFF command

The Option panels allow you to select runtime options. CA JCLCheck saves these options in your ISPF profile for subsequent requests. You can also access help using the Help key.

If you use CA Roscoe rather than ISPF, turn on the CA7CTL option.

Note: For information about the invocation of CA JCLCheck and the ISPF dialog, see Install JCLNeat ISPF Interface in the *Installation Guide*.

You can use the CA JCLCheck ISPF dialog for:

- Accessing the CA JCLCheck SPF Menu (JCK0200) (see page 17)
- Setting up CA JCLCheck options (see page 18)
- Setting up your job card (see page 20)
- Viewing options through CA JCLCheck invocation panels (see page 22)
- Scanning a JCL member (see page 23)
- Scanning a group of JCL members (see page 24)
- Specifying CA JCLCheck options through the invocation panels (see page 25)
- Submitting JCL to run CA JCLCheck in batch (see page 26)

If you are a new CA JCLCheck user, you should perform all of the above instructions associated with the ISPF dialog.

Accessing the CA JCLCheck SPF Menu (JCK0200)

To access the CA JCLCheck SPF menu, do one of the following:

- Enter **%JCKSPF** from TSO or ISPF 6.
- Enter **J** to select CA JCLCheck from the ISPF menu, if installed.

Press enter.

```
 -------------  ISPF/PDF  PRIMARY  OPTION  MENU  ---------------------

 OPTION      ===>TSO %JCKSPF

                                                        USERID   - USER02
0 ISPF PARMS   - Specify terminal and user parameters   TIME     - 10:57
1 BROWSE       - Display source data or output listings TERMINAL - 3278
2 EDIT         - Create or change source data           PF KEYS  - 12
3 UTILITIES    - Perform utility functions
4 FOREGROUND   - Invoke language processors in foreground
5 BATCH        - Submit job for language processing
6 COMMAND      - Enter TSO command or CLIST
7 DIALOG TEST  - Perform dialog testing
8 LM UTILITIES - Perform library administrator utility functions
9 IBM PRODUCTS - Additional IBM program development products
C CHANGES      - Display summary of changes for this release
T TUTORIAL     - Display information about ISPF/PDF
X EXIT         - Terminate ISPF using log and list defaults
J JCLCHECK     - CA JCLCHECK ISPF Interface

 Enter End command to terminate ISPF.
```

Setting up CA JCLCheck Options

1. From the CA JCLCheck SPF Menu, enter **0** on the OPTION line to select Option Specifications, press enter.
2. From the CA JCLCheck Control Options Menu, select **1**, **2**, or **3** to change your Process Control Options, Output Control Options, or Security Options, press enter.

   Note: For more information about specifying output control options using the Output Control Options panel, see Selecting Reports (see page 27). For more information about specifying security options using the Security Option panel, see Defining Syntax Checking and Message Severity (see page 42).
3. From the CA JCLCheck Process Control Options, Output Control Options, or Security Options Menu, enter **Y** in the Update PROFILE with these options field to save the changes, press enter.
4. Press **F3** (End) to return to the previous panel (Control Options Menu).

```
JCK0200-------------- CA JCLCHECK/SPF MENU ------------------------
OPTION ===> 0


  Date:  yyyy/mm/dd    User: USERID1    Level: xx.x

       0  -  Option Specifications

       1  -  CA JCLCHECK - Foreground Invocation

       2  -  CA JCLCHECK - Foreground Invocation / Submit

       3  -  CA JCLCHECK - Batch Submit

       4  -  CA 7 Interface Invocation

       5  -  CA Scheduler Interface Invocation

       6  -  JCLNeat - Batch Submit

       7  -  RESOLVE - External Variable Resolution

       X  -  Exit
```

```
  ------------- CA JCLCHECK/CONTROL OPTIONS MENU ---------------------
  OPTION ===> 1


  Date:  yyyy/mm/dd    User: USER02    Time: 10:58

       1  -  Process Control Options

       2  -  Output Control Options

       3  -  Security Options

       4  -  JCLNeat Options

       5  -  Job Card Specifications

       X  -  Exit
```

```
------------------- CA JCLCheck PROCESS CONTROL OPTIONS -------------------
COMMAND ===>
 Date: yyyy/mm/dd               User: USERID1                    Time: hh:mm

           Go to ADDITIONAL PROCESSING OPTIONS PANEL   => N

CCLIST   - Number of program control cards to be listed       (0-999) => 100
COND     - Simulate execution according to condition codes            => N
CTLSCAN  - Scan utility control statements                            => N
EASYPROC - CA EASYPROCLIB support                                     => N
MSS      - Stage VTOC of MSS volumes if necessary                     => N
PXREF    - Check program and PDS member existence                     => N
SEV      - Minimum severity of errors to be listed             (0-16) => 0
SPACE    - Check blksize efficiency for new DASD files                => N
   (ALL) - Check blksize efficiency for existing DASD files too       => N
   (NN)  - Minimum % of track usage (give warning if less)     (0-99) => 65
SXREF    - Check JCL across jobs                                      => N
VSAM     - Process private catalogs (JOBCAT, STEPCAT)                 => Y
CSI      - Use Catalog Search Interface for catalog lookup            => D
PROC     - Define alternate proclibs                                  => N
         - Alternate proclib(s) are selected                          => Y
DICT     - Define CA APCDOC XREF database                             => N

           Update PROFILE with these options          => Y
```

Setting up your job card

1. From the Control Options menu, enter **5** on the OPTION line to select Job Card Specifications, press enter.
2. From the Specification of Job Card menu, specify the job statement you want CA JCLCheck to use when submitting a job in batch.
3. Enter **Y** in the Update PROFILE with these options field to save the new job statement, press enter.
4. Press **F3** (End) to return to the previous panel (Control Options Menu).

   When you finish defining your options and job statement, press **F3** (End) one more time to return to the SPF menu.

```
 ------------------- CA JCLCHECK/CONTROL OPTIONS MENU ---------------
 OPTION ===> 5

 Date:  yyyy/mm/dd    User: USER02    Time: 10:58

      1  -  Process Control Options

      2  -  Output Control Options

      3  -  Security Options

      4  -  JCLNeat Options

      5  -  Job Card Specifications

      X  -  Exit
```

```
---------------------- SPECIFICATION OF JOB CARD ----------------------
 COMMAND===>

        Date:  yyyy/mm/dd       Userid: USER02   Time: 10:58

        Job Statement Information:  (Verify before submitting)

  ===> //USER02H JOB (40900000),'LINDA',MSGCLASS=X,CLASS=K,
//         USER=USER02
/*ROUTE PRINT SYSTEMC.USER02
//*


Update PROFILE with these options                          => Y
```

Viewing options through CA JCLCheck invocation panels

1. Enter **1**, **2**, or **3** (Foreground Invocation, Foreground Invocation Submit, Batch Submit), press enter.

   CA JCLCheck displays the options that you specified in the OPTIONS fields at the bottom of the INVOCATION panel.

```
JCK0200-------------- CA JCLCHECK/SPF MENU ------------------------
OPTION ===> 1


  Date:  yyyy/mm/dd     User: USERID1    Level: xx.x

       0  -  Option Specifications

       1  -  CA JCLCHECK - Foreground Invocation

       2  -  CA JCLCHECK - Foreground Invocation / Submit

       3  -  CA JCLCHECK - Batch Submit

       4  -  CA 7 Interface Invocation

       5  -  CA Scheduler Interface Invocation

       6  -  JCLNeat - Batch Submit

       7  -  RESOLVE - External Variable Resolution

       X  -  Exit
```

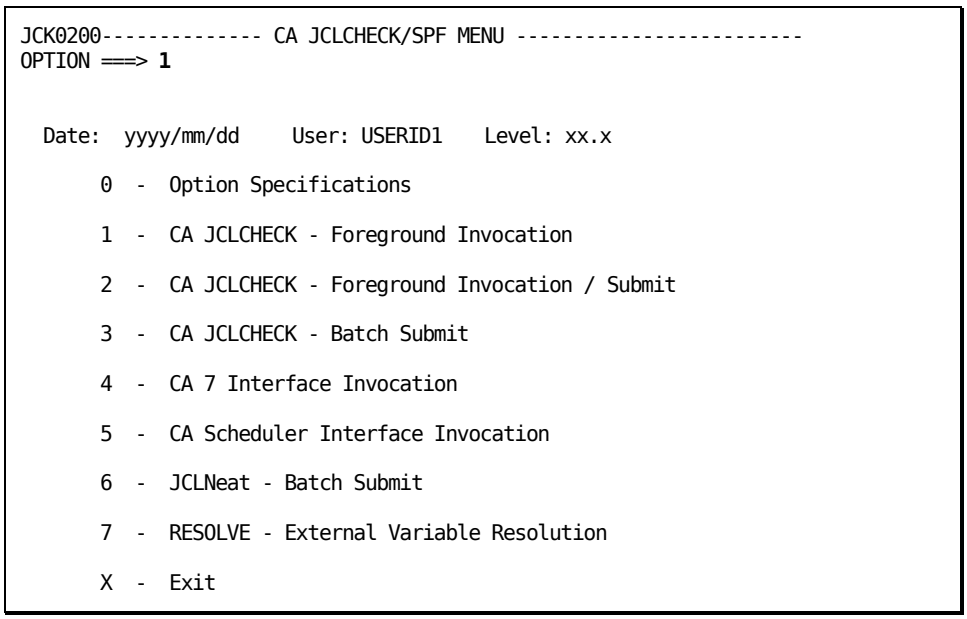
```
  ------------------------ CA JCLCHECK/INVOCATION ---------------------
  COMMAND ===>

  Option Selected: FOREGROUND

   Date:  yyyy/mm/dd            User: USER02                      Time: 10:59

     Data Set Name      ===> 'CAI.APCMTL.CHQAJCL'
     Library Type       ===> PDS        (SEQ, PDS, PAN, LIB)
     Member             ===> USER02     (Batch Submit -- Use * for all members)
     Volume Serial      ===>           (If Not CATALOGED)
     CPU-ID             ===>           (Multi-CPU Only, blank: This CPU)
     SYSTERM            ===> N       (Y/N/P, Error Statement and Messages only)
     SYSPRINT           ===> Y       (Y/N/P, Reports 1-9 according to options )

     Specify any additional options (FOR THIS RUN ONLY) below
     OPTIONS    ===> CC(5)  CT    SP(RPT 65 NOALL) V
     OPTIONS    ===> J F XREF PXR(RPT) SX RP ER( END)
     OPTIONS    ===> SYN RUNT
     OPTIONS    ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
     OPTIONS    ===>
     SAVED      ===>
     SAVED      ===>
```

Note: You can also enter options directly into the OPTIONS and SAVED fields on the INVOCATION panel. If you enter options in the OPTIONS fields, the options are only active for the current run of CA JCLCheck. If you enter options in the SAVED field, the options are saved for multiple runs.

Enter **Y** (yes), **N** (no), or **P** (print) in the SYSTERM and SYSPRINT fields on the CA JCLCheck INVOCATION panel, depending on whether you want to view or print error statement and messages, and reports.

Scanning a JCL member

1. From the SPF Menu, enter **1** (Foreground Invocation), press enter.
2. From the Invocation menu, enter a data set name, library type, and member name, press enter. This invokes CA JCLCheck to scan the selected member.



```
------------------ CA JCLCHECK/INVOCATION -----------------------------
COMMAND ===>

Option Selected: FOREGROUND

 Date:  yyyy/mm/dd            User: USER02             Time: 10:59

   Data Set Name      ===> 'CAI.APCMTL.CHQAJCL'
   Library Type       ===> PDS      (SEQ, PDS, PAN, LIB)
   Member             ===> USER02   (Batch Submit -- Use * for all members)
   Volume Serial      ===>    (If Not CATALOGED)
   CPU-ID             ===>    (Multi-CPU Only, blank: This CPU)
   SYSTERM            ===> N  (Y/N/P, Error Statement and Messages only)
   SYSPRINT           ===> Y  (Y/N/P, Reports 1-9 according to options )

   Specify any additional options (FOR THIS RUN ONLY) below
   OPTIONS            ===> CC(5)  CT    SP(RPT 65 NOALL) V
   OPTIONS            ===> J F XREF PXR(RPT) SX RP ER( END)
   OPTIONS            ===> SYN RUNT
   OPTIONS            ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
   OPTIONS            ===>
   SAVED              ===>
   SAVED              ===>
```

CA JCLCheck displays a series of reports, beginning with a small informational report that displays the calling parameters, options, and installation defaults (from JCLDFLT) used for this run. The main purpose of this report is to display the options in effect. The rest of the CA JCLCheck reports are specified by the options you select.

Scanning a group of JCL members

1. From the SPF Menu, enter **1** (Foreground Invocation), press enter.
2. From the Invocation menu, enter a data set name and library type, press enter.

   CA JCLCheck presents you with a member selection list.
3. To determine the sequence that members should be scanned, enter a **sequence number** in the CMD column beside the name.
4. To add a member to the bottom of the list, put an **N** (Next) in the CMD column, press enter.

   The selected members move above a divider line (========), in the order specified by the numbering sequence.
5. If you want to deselect a selected member, enter **0** beside the name (above the divider line), press enter.

   Press **F3** (End) to run CA JCLCheck against these members.

```
 ----------------  CA JCLCHECK MEMBER SELECT  ------------  ROW 1 OF 14
COMMAND ===>                                              SCROLL ===> PAGE

  Data Set Name: CAI.APCMTL.CHQAJCL                       Type .: PDS

CMD   MEMBER   ORDER   CREATED          LAST MODIFIED    SIZE    USERID
      BACKUP           2000/09/10       2000/09/20  08:29        30      USER02
      CAZ1JCHK         2000/08/19       2000/08/20  09:07        20      USER02
      IEBCOPY          2000/08/19       2000/08/19  16:18        12      E287
 1    USER02           2000/08/19       2000/08/19  16:50        19      E287
 2    QAREXX           2000/08/19       2000/08/20  08:49        19      USER02
 3    RESTORE          2000/09/16       2000/09/16  15:11        26      USER02
 N    ZXJCSCMP         2000/09/10       2000/09/13  08:50        65      USER02
 ************************** BOTTOM OF DATA ***********************************
```

```
 ----------------  CA JCLCHECK MEMBER SELECT  -----------  ROW 1 OF 14
COMMAND ===>                                              SCROLL ===> PAGE

  Data Set Name: CAI.APCMTL.CHQAJCL                                 Type .: PDS

CMD   MEMBER   ORDER   CREATED          LAST MODIFIED      SIZE  USERID
      USER02     1     2000/08/19       2000/08/19  16:50  19    E287
 0    QAREXX     2     2000/08/19       2000/08/20  08:49  19    USER02
      RESTORE    3     2000/09/16       2000/09/16  15:11  26    USER02
      ZXJCSCMP   4     2000/09/10       2000/09/13  08:50  65    USER02
     ========   ====   ========         ======== =====   ====    =======
      BACKUP           2000/09/10       2000/09/20  08:29  30    USER02
      CAZ1JCHK         2000/08/19       2000/08/20  09:07  20    USER02
      IEBCOPY          2000/08/19       2000/08/19  16:18  12    E287
 ************************ BOTTOM OF DATA ***********************************
```

Specifying CA JCLCheck options through the invocation panels

1. From the Invocation menu, enter a data set name, library type, and insert **PDSR** and **SXREF** in the OPTIONS fields, press enter.

   CA JCLCheck presents you with a member selection list.

2. To determine the sequence that members should be scanned, enter a **sequence number** in the CMD column beside the name.

3. To add a member to the bottom of the list, put an **N** (Next) in the CMD column, press enter.

   The selected members move above a divider line (========), in the order specified by the numbering sequence.

4. If you want to deselect a selected member, enter **0** beside the name (above the divider line), press enter.

5. Press **F3** (End).

```
------------------- CA JCLCHECK/INVOCATION --------------------------
COMMAND ===>

Option Selected: FOREGROUND

 Date:  yyyy/mm/dd           User: USER02                    Time: 10:59

   Data Set Name      ===> 'CAI.APCMTL.CHQAJCL'
   Library Type       ===> PDS      (SEQ, PDS, PAN, LIB)
   Member             ===> USER02  (Batch Submit -- Use * for all members)
   Volume Serial      ===>         (If Not CATALOGED)
   CPU-ID             ===>         (Multi-CPU Only, blank: This CPU)
   SYSTERM            ===>         (Y/N/P, Error Statement and Messages only)
   SYSPRINT           ===> Y       (Y/N/P, Reports 1-9 according to options )


   Specify any additional options (FOR THIS RUN ONLY) below
   OPTIONS    ===> CC(5)  CT    SP(RPT 65 NOALL) V
   OPTIONS    ===> J F XREF PXR(RPT) SX RP ER( END)
   OPTIONS    ===> SYN RUNT PDSR SXREF
   OPTIONS    ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
   OPTIONS    ===>
   SAVED      ===>
   SAVED      ===>
```

This causes CA JCLCheck to treat the jobs in these members as a series of consecutive, related job streams. There are dependencies from job to job. If a job deletes a data set and another job in that run has DISP=SHR, referring to the deleted data set, you receive an error in Report 6 - ERROR MESSAGES.

Submitting JCL to run CA JCLCheck in batch

1. From the SPF Menu, enter **3** (Batch Submit), press enter.
2. From the Invocation menu, enter a data set name, library type, and insert **SXREF** and **PROCXREF** in the OPTIONS fields, press enter.

   CA JCLCheck presents you with a member selection list.
3. To determine the sequence that members should be scanned, enter a **sequence number** in the CMD column beside the name.
4. To add a member to the bottom of the list, put an **N** (Next) in the CMD column, press enter.

   The selected members move above a divider line (========), in the order specified by the numbering sequence.
5. If you want to deselect a selected member, enter **0** beside the name (above the divider line), press enter.

6. When you complete your selections, press **F3** (End).

   CA JCLCheck submits a job. View the generated reports from the SDSF option available from the ISPF menu (if the output was directed to the TSO hold queue).

Note: For information about the use of the ISPF dialog, see Using CA JCLCheck in the *Command Reference Guide*. For information about runtime options, see CA JCLCheck Runtime Options in the *Command Reference Guide*.

Selecting Reports

The CA JCLCheck reporting system documents submitted JCL for a single job or for a full production job stream, including data set and program cross-references, and error message listings. Viewing these reports reduces production problems by having factual, easily-obtainable JCL listings prior to job submission. The reports provide the Systems Operator with an opportunity to correct error situations before running the jobs.

If you specify the GRAPH option, the JCL is charted in standard job flow format, suitable for use in a runbook.

If you specify the AUDIT option, you will produce an Audit Report with a statistical summary, input records, and averages of all related jobs, steps, and DD statements.

In this section you are provided with instructions to produce Report 1 - Listing of Jobstream JCL, using the JOB option. To produce all subsequent reports, use the instructions from Report 1, the report option for the new report, and any other implementation information provided for that report.

You can select Options 1, 2, or 3 (Foreground, Foreground Submit, Batch Submit) from the CA JCLCheck/SPF Menu to run CA JCLCheck against JCL to produce CA JCLCheck reports. Option 1 (Foreground Invocation) is used in the example.

Note: For more information about reports, see the *Command Reference Guide*.

This section shows you how to produce the following reports:

- Report 1 - Listing of Jobstream JCL (see page 28)
- Report 2 - Listing of Merged JCL (see page 31)
- Report 3 - Data Set Cross-Reference (see page 32)
- Report 4 - Program Cross-Reference (see page 32)
- Report 5 - Report Listing (see page 33)
- Report 6 - Error Messages (see page 33)

- Report 7 - Summary Data Set Cross-Reference (see page 34)
- Report 8 - Summary Program Cross-Reference (see page 35)
- Report 9 - Summary Report Listing (see page 36)
- Report 10 - Flow Diagram (see page 36)
- Report 11 - Procedure Cross-Reference (see page 40)
- Audit Report (see page 41)

Report 1 - Listing of Jobstream JCL

The Listing of Jobstream JCL report is a card image listing of the JCL as submitted. The report also includes control statements if the CCLIST option is requested.

Use the following steps to view the various types of CA JCLCheck reports, substituting the option of the report you want to produce.

Note: Depending on which report options may already be active, each step may produce any series of reports.

1. From the CA JCLCheck SPF Menu, enter **0** on the OPTION line to select Option Specifications, press enter.
2. From the CA JCLCheck Control Options Menu, enter **2** to select Output Control Options, press enter.

3. From the CA JCLCheck Output Control Options panel, enter **Y** in the JOB field and press **F3** (End). Alternatively, you can enter the JOB option in the OPTIONS field on the INVOCATION panel, and press enter to run CA JCLCheck against a JCL member.

```
JCK0200-------------- CA JCLCHECK/SPF MENU ------------------------
OPTION ===> 0


  Date:  yyyy/mm/dd     User: USERID1    Level: xx.x

       0  -  Option Specifications

       1  -  CA JCLCHECK - Foreground Invocation

       2  -  CA JCLCHECK - Foreground Invocation / Submit

       3  -  CA JCLCHECK - Batch Submit

       4  -  CA 7 Interface Invocation

       5  -  CA Scheduler Interface Invocation

       6  -  JCLNeat - Batch Submit

       7  -  RESOLVE - External Variable Resolution

       X  -  Exit
```

```
   ------------- CA JCLCHECK/CONTROL OPTIONS MENU ----------------------
  OPTION ===> 2

   Date:  yyyy/mm/dd            User: USER02             Time: 10:58

       1  -  Process Control Options

       2  -  Output Control Options

       3  -  Security Options

       4  -  JCLNeat Options

       5  -  Job Card Specifications

       X  -  Exit
```

```
 -------------------- JCLCHECK OUTPUT CONTROL OPTIONS --------------------
   COMMAND ===>

    Date:  yyyy/mm/dd          User: USER02                      Time: 16:30

   AUDIT       - Audit Report producing totals and averages              => Y
   JOB         - List input JCL                                 (RPT 1)  => Y
   FULLLIST    - List complete merged JCL and procedures        (RPT 2)  => Y
   XREF        - Produce Data Set Cross-Reference       (RPT 3)          => Y
   PXREF       - Produce Program Cross-Reference                (RPT 4)  => Y
   RPTRPT      - List SYSOUT Reports                            (RPT 5)  => Y
   ERROR(END)  - Place Error Report last                (RPT 6)  => Y
      (START)  - Place Error Report first                                 => N
     (INLINE)  - Place Error Messages INLINE in Report 2                  => N

   SPACE(RPT)  - List DASD Space in Dataset Reports 3,7                   => Y
   SX(RPT)     - Summary DSN, PGM, RPT Cross-Reference   (RPT 7,8,9)      => Y
   SX(ONLY)    - Summary RPTS 7,8,9 only(SUPPRESS RPT 3,4,5)     => N
   PROCXREF    - Symbolic Reference Report              (RPT 11)         => N
   GRAPH       - Produce JOB Flowcharts                 (RPT 10)         => N
               - Print the JOB Flowcharts                                 => N
   MCOUSR      - Message Control Option - User Level DDNAME               => N
                              => MCOUSR
   MCOSYS      - Message Control Option - System Level DDNAME             => D
                              => MEMBER=

               Update Profile With These Options                          => Y
```

```
---------------------CA JCLCHECK/INVOCATION -----------------------------
  COMMAND ===>

  Option Selected: FOREGROUND

   Date:  yyyy/mm/dd            User: USER02             Time: 10:59

     Data Set Name      ===> 'CAI.APCMTL.CHQAJCL'
     Library Type       ===> PDS         (SEQ, PDS, PAN, LIB)
     Member             ===> Z1JCSCMP (Batch Submit -- Use * for all members)
     Volume Serial      ===>           (If Not CATALOGED)
     CPU-ID             ===>           (Multi-CPU Only, blank: This CPU)
     SYSTERM            ===> N         (Y/N/P, Error Statement and Messages only)
     SYSPRINT           ===> Y         (Y/N/P, Reports 1-9 according to options )

     Specify any additional options (FOR THIS RUN ONLY) below
     OPTIONS    ===> CC(5)  CT    SP(RPT 65 NOALL) V
     OPTIONS    ===> JOB F XREF PXR(RPT) SX RP ER( END)
     OPTIONS    ===> SYN RUNT
     OPTIONS    ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
     OPTIONS    ===>
     SAVED      ===>
     SAVED      ===>
```

Sample Report 1 - Listing of Jobstream JCL

```
CA                                  REPORT 1 - LISTING OF JOBSTREAM JCL
CA JCLCHECK             SYS: xxxx     JOB: USER002         PGMR: JOB CONTROL COMPILE                THURSDAY JANUARY 9, yyyy
-------------------------------------------------------------------------------------------------------------------------------
STMT                                >>> INPUT DSNAME = PROD.JCL.LIBRARY                                          <<<
NUM  STATEMENT TEXT                                                      >>> INPUT MEMBER = USER002      <<<

  1. //USER002 JOB (41400000,IGN),'JOB CONTROL COMPILE',CLASS=K,MSGCLASS=X
  2. //*Z1JCSCMP JOB ...
  3. //*
  4. //*    THIS IS A MODIFICATION ON Z1JCSCMP FROM JCLCHECK 7.1
  5. //*    THE THREE CATALOGED PROCEDURES HAVE BEEN MADE INLINE TO
  6. //*    FACILITATE MODIFICATIONS.
  7. //*
  8. //*********************************************************************************************
  9. //* COMPILE, ASSEMBLE, AND LINK JOB CONTROL STANDARDS MODULE CAZ1SUSR
 10. //* ---------------------------------------------------------------------------------------------------------------------
 11. //* THIS JOB WILL COMPILE JOB CONTROL STANDARDS (JCS) THAT HAVE BEEN
 12. //* DEFINED IN THE CA JCLCHECK JCS RULES DATABASE. THE RESULT OF THE
 13. //* COMPILE IS A SERIES OF MACROS THAT ARE ASSEMBLED AND LINKED AS
 14. //* MODULE 'CAZ1SUSR'. THIS MODULE IS CALLED WHENEVER THE CA JCLCHECK
 15. //* 'STANDARD' OPTION REQUESTING JCS ENFORCEMENT IS SPECIFIED AT
 16. //* RUNTIME.
 17. //*********************************************************************************************
 18. //*
 19. //* ---------------------------------------------------------------------------------------------------------------------
 20. //* PROCEDURE TO COMPILE JOB CONTROL STANDARDS FROM THE JCLCHECK
```

Report 2 - Listing of Merged JCL

The Listing of Merged JCL report is a listing of the entire JCL for the job, including procedure library statements. For ease of use, the listing is similar to pre-z/OS format; procedure statements and substitution JCL statements are listed in line, but are indented four columns. Based on the setting of the FULLLIST option, this report contains either all of the job's JCL or only those statements flagged with error messages.

Follow the instructions for Report 1 - LISTING OF JOBSTREAM JCL and specify the **FULLLIST** option.

Sample Report 2 - Listing of Merged JCL

```
CA                                  REPORT 2 - LISTING OF MERGED JCL
CA JCLCHECK             SYS: xxxx     JOB: USER002         PGMR: JOB CONTROL COMPILE                THURSDAY JANUARY 9, yyyy
-------------------------------------------------------------------------------------------------------------------------------
  STMT  STEP      PROCSTEP   ERR      CMNT        >>> INPUT DSNAME = PROD.JCL.LIBRARY                          <<<
  NUM   NAME      NAME       SEV       COL        STATEMENT TEXT                 >>> INPUT MEMBER = USER002      <<<
-------------------------------------------------------------------------------------------------------------------------------
   1.                        12                   //USER002 JOB (41400000,IGN),'JOB CONTROL C
   2.                                             //*Z1JCSCMP JOB ...
   3.                                             //*
   4.                                             //*    THIS IS A MODIFICATION ON Z1JCSCMP FR
   5.                                             //*    THE THREE CATALOGED PROCEDURES HAVE B
   6.                                             //*    FACILITATE MODIFICATIONS.
   7.                                             //*
   8.                                             //*****************************************
   9.                                             //* COMPILE, ASSEMBLE, AND LINK JOB CONTROL
```

```
14.                                     //*
15.                                     //*************************************
16.                                     //* PROCEDURE CAZ1JCS
17.                                     //* --------------------------------
18.                                     //* PROCEDURE TO COMPILE JOB CONTROL STA
19.                                     //* JOB CONTROL STANDARDS (JCS) RULES DA
20.                                     //* COMPILE CONSISTS OF A SERIES OF MACR
21.                                     //* AND LINKED AS MODULE 'CAZ1SUSR'.
22.                                     //*************************************
23.                                     //*
```

Report 3 - Data Set Cross-Reference

The Data Set Cross-Reference report lists all data sets and members used by the job, and that exist for more than one step. Included in the report are all permanent data sets and all temporary data sets with a disposition other than NEW, DELETE.

Follow the instructions for Report 1 - LISTING OF JOBSTREAM JCL and specify the **XREF** option.

Sample Report 3 - Data Set Cross-Reference

```
CA                                  REPORT 3 - DATA SET CROSS REFERENCE                                  PAGE        3
CA JCLCHECK             SYS: xxxx       JOB: BK042005     PGMR: CAUSER02                 WEDNESDAY SEPTEMBER 12, yyyy
-------------------------------------------------------------------------------------------------------------------------
DATA SET NAME                           MEMBER            DSET    DCB INFO      DEFN    DELETE    CATLG  REFERENCED
                              SPACE              VOL(S)  TYPE  FMT LRECL,BLK   STMT    STMT      STMT   IN STATEMENTS
-------------------------------------------------------------------------------------------------------------------------

KNEBO01.$J6ESTG.LISTING                          >TSU028  PDSE  F 133,3325
                                                                                                        3
                                        DATACLAS: DCWRKD    STORCLAS: TSO       MGMTCLAS: TSO
KNEBO01.MYLIB.N                          ??????
                                        ASMDSECT                                                        8
                                        CONVDATE                                                        9
                                        CONVTIME                                                        6
                                                                                                        11
                                        IEBGENER                                                        4
                                                                                                        5
KNEBO01.REP.VOLSER2                       ??????
                                                                                                        12
KNEBO01.REPORT#3                         >TSU014  PDSE  F 80,3120
                                        ASM#ONLY                                                        7
                                        DATACLAS: DEFAULT   STORCLAS: TSO       MGMTCLAS: TSO
*************************************************************************************************************************
```

Report 4 - Program Cross-Reference

The Program Cross-Reference report lists all programs used by the job and the libraries where they reside. This includes all programs referenced on EXEC statements and programs invoked by IMS (specified in the second PARM field parameter when the program name is DFSRRC00).

Follow the instructions for LISTING OF JOBSTREAM JCL and specify the **PXREF** option.

Sample Report 4 - Program Cross-Reference

```
CA                                  REPORT 4 - PROGRAM CROSS REFERENCE
CA JCLCHECK             SYS:        JOB: JCHKTEST       PGMR: CAI-JB0                          THURSDAY JANUARY 9, yyyy
-------------------------------------------------------------------------------------------------------------------------------
PROGRAM          LIBRARY    CONTAINING  PROGRAM                                  REFERENCED IN
NAME             VOLUME     DSNAME                                               STATEMENT (STEP)
-------------------------------------------------------------------------------------------------------------------------------

IEBGENER         MVXAD2     SYS1.LINKLIB                                          15 (STEP12.NSTEP2)
IEFBR14          SYS1.LINKLIB                                        8 (PROC.STEP1) 13 (STEP12.NSTEP1)
```

Report 5 - Report Listing

The Report Listing contains all SYSOUT reports produced by the job. A SYSOUT report is assumed to result from any DD statement specifying the SYSOUT parameter, including those requesting an internal reader. Information for this report is obtained from both the SYSOUT DD statement itself and from any associated JES2 /*OUTPUT control statement.

Follow the instructions for REPORT 1 - LISTING OF JOBSTREAM JCL and specify the **RPTRPT** option.

Sample Report 5 - Report Listing

```
CA                                  REPORT 5 - REPORT REPORT
CA JCLCHECK             SYS: xxxx   JOB: USER002       PGMR: JOB CONTROL COMPILE              THURSDAY JANUARY 9, yyyy
-------------------------------------------------------------------------------------------------------------------------------
  STMT  STEP      PROCSTEP          SYSOUT    SPECIAL   COPY      SYSOUT    CA DISPATCH        REPORT
  NUM   NAME      NAME      DDNAME  CLASS     FORMS     COUNT     PROGRAM                      DESTINATION(S)
-------------------------------------------------------------------------------------------------------------------------------

  29.                       SYSOUT      X
  34.   ASM1                SYSPRINT    X
  50.   LINK1               SYSPRINT    X
```

Report 6 - Error Messages

The Error Messages report lists all error messages associated with this job. It is produced only if one or more errors are detected. Direct this report to appear at the beginning or end of the report output or incorporate it with Report 2 - Listing of Merged JCL, depending on the setting of the ERROR option.

Follow the instructions for Report 1 - LISTING OF JOBSTREAM JCL and specify the **ERROR** option.

Do one of the following:

- Enter **Y** in the ERROR(END) field on the Output Control Options panel. Press **F3** (End) to exit the panel.
- Enter **ER(S), ER(E), ER(I), ER(S I)**, or **ER(I E)** in the OPTIONS field on one of the invocation panels and run CA JCLCheck against a JCL member. View the results.

Enter	To have CA JCLCheck:
ER(S)	Start REPORT 6 - ERROR MESSAGES before all other reports.
ER(E)	Put the Error Messages report last.
ER(I)	Embed error messages within Report 2 (your listing of merged JCL).
ER(S I)	Display Report 2 first with the error messages embedded.
ER(E I)	Display Report 2 last with error messages embedded.

Sample Report 6 - Error Messages

```
CA                                  REPORT 6 - ERROR MESSAGES
CA JCLCHECK             SYS: xxxx     JOB: USER002         PGMR: JOB CONTROL COMPILE                THURSDAY JANUARY 9, yyyy
-------------------------------------------------------------------------------------------------------------------------------
  STMT  MESSAGE                                  >>> INPUT DSNAME = PROD.JCL.LIBRARY                              <<<
  NUM   NUMBER    MESSAGE TEXT                                                >>> INPUT MEMBER = USER002      <<<
-------------------------------------------------------------------------------------------------------------------------------
   1.   CAY6326S   'VSIGNON' UNKNOWN RETURN CODE 16 FROM CAISSF, SECURITY VALIDAT
        CAY6323I   FEEDBACK= 'CAS9TSS: UNKNOWN FUNCTION NOT SUPPORTED'
        CAY6183E   GRAPH OUTPUT TERMINATED DUE TO ERRORS IN THE JCL
  25.   CAY6025E   REQUIRED PGM OR PROC PARAMETER MISSING
        CAY6037E   UNKNOWN KEYWORD 'PPM' SPECIFIED IN OPERAND OF 'EXEC'
  27.   CAY6077W   NO '3050200E' UNITS ARE AVAILABLE FOR 'ARCHIV'
        CAY6067W   UNABLE TO VERIFY THAT DATA SET 'CAI.APCMTL.JCLCHK.STDRULE
  33.   CAY6051E   STATEMENT REFERENCED BY BACK REFERENCE 'EVAL1' CANNOT BE FOUND
        CAY6145I   CONDITION CODE PROCESSING TERMINATED DUE TO ERRORS
  38.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S0
CAY6000 5 STATEMENTS FLAGGED IN JOB 'USER002' MAXIMUM SEVERITY WAS 12
```

Report 7 - Summary Data Set Cross-Reference

The Summary Data Set Cross-Reference report lists all permanent data sets and members used by any job within a production system (series of jobs analyzed using a single run of CA JCLCheck). It is produced only if both the SXREF and XREF options are selected. The report is similar to Report 3 - Data Set Cross-Reference, except that all data sets used by the related jobs are included. The report is sequenced by data set name.

Follow the instructions for Report 1 - LISTING OF JOBSTREAM JCL and specify two options: **SX(RPT)** and **SX(ONLY)**.

Sample Report 7 - Summary Data Set Cross-Reference

```
CA                                  REPORT 7 - SUMMARY DATA SET CROSS REFERENCE                                  PAGE      1
CA JCLCHECK             SYS: xxxx                                                                  WEDNESDAY SEPTEMBER 12, yyyy
-------------------------------------------------------------------------------------------------------------------------------
DATA SET NAME                          MEMBER            DSET    DCB INFO    REFERENCED IN
                                 SPACE               VOL(S)  TYPE  FMT LRECL,BLK JOB (STATUS)  - STEP  (STMT)
-------------------------------------------------------------------------------------------------------------------------------
KNEBO01.$J6ESTG.LISTING                            >TSU028  PDSE  F 133,3325    BK042005 -
                                                                                       S1.DDD00 (3)
KNEBO01.MYLIB.N                                     ??????                     BK042005 -
                                         ASMDSECT                                      S1.DDD3 (8)
                                         CONVDATE                                      S1.DDD4 (9)
                                         CONVTIME                                      S1.DDD1 (6)
                                                                                       S2.DDD2 (11)
                                         IEBGENER                                      S1.DDD1A (4)
                                                                                       S1.DDD1B (5)
KNEBO01.REP.VOLSER2                                 ??????                     BK042005 -
                                                                                       S2.DDDX (12)
KNEBO01.REPORT#3                                   >TSU014  PDSE  F 80,3120     BK042005 -
                                         ASM#ONLY                                      S1.DDDA (7)
*******************************************************************************************************************************
```

Report 8 - Summary Program Cross-Reference

The Summary Program Cross-Reference report lists all programs used by any job within a production system (series of jobs analyzed using a single run of CA JCLCheck). The report is similar to the Report 4 - Program Cross-Reference, except that all programs included in the system are reported in program name sequence. The listing includes all programs referenced on EXEC statements and programs invoked by IMS (specified in the second PARM field parameter when the program name is DFSRRC00).

Follow the instructions for Report 1 - LISTING OF JOBSTREAM JCL and specify two options: **SXREF** and **PXREF**.

Sample Report 8 - Summary Program Cross-Reference

```
CA                                  REPORT 8 - SUMMARY PROGRAM CROSS REFERENCE
CA JCLCHECK             SYS: xxxx                                                  THURSDAY JANUARY 9, yyyy
-------------------------------------------------------------------------------------------------------------------------------
PROGRAM    LIBRARY  CONTAINING PROGRAM                              REFERENCED IN
NAME       VOLUME   DSNAME                                         JOB, STEP (STATEMENT)
-------------------------------------------------------------------------------------------------------------------------------
IEBGENER   MVXAD2   SYS1.LINKLIB                                   JCHKTEST, STEP12.NSTEP2 (15)
IEFBR14             SYS1.LINKLIB                                   JCHKTEST, PROC.STEP1 (8) JCHKTEST, STEP12.N
```

Report 9 – Summary Report Listing

The Summary Report Listing contains all SYSOUT reports produced by any job within a production system (series of jobs analyzed using a single run of CA JCLCheck). It is produced only if the SXREF and RPTRPT options are both selected. A SYSOUT report is assumed to result from any DD statement specifying the SYSOUT parameter, including those requesting an internal reader. Information for this report is obtained from both the SYSOUT DD statement itself and from any associated JES2 /*OUTPUT control statement.

Follow the instructions for Report 1 - LISTING OF JOBSTREAM JCL and specify two options: **RPTRPT** and **SXREF** .

Sample Report 9 - Summary Report Listing

```
CA                                  REPORT 9 - SUMMARY REPORT REPORT
CA JCLCHECK              SYS: xxxx                                                                    THURSDAY JANUARY 9, yyyy
-------------------------------------------------------------------------------------------------------------------------------
 STMT   JOB       PROCSTEP  STEP              SYSOUT    SPECIAL   COPY      SYSOUT    REPORT
 NUM    NAME      NAME      NAME     DDNAME   CLASS     FORMS     COUNT     PROGRAM   DESTINATION
-------------------------------------------------------------------------------------------------------------------------------
 18.    JCHKTEST  STEP12    NSTEP2   SYSPRINT    J      0004        4       ________  TSC32622
```

Report 10 - Flow Diagram

The Flow Diagram report is a graphic representation of the flow of the JCL processed by CA JCLCheck. The first page lists the name of the JCL member being charted and any JOBLIB, JOBCAT, or PROCLIB data sets associated with the job. STEPLIB and STEPCAT data sets are listed at the beginning of each step. Input and output data sets, respectively, are listed on the left and right side of the process box (procstep/step/program). As much relevant information as is available from the JCL or the system catalog is also listed for each symbol:

- Data set name
- Data set disposition
- Unit and volume information (all volumes)
- Data set attributes: LRECL, BLKSIZE, DSORG, RECFM

In addition, this information is listed for tape data sets:

- LABEL type
- File number

In addition, this information is listed for SYSOUT data sets:

- SYSOUT CLASS
- FCB/UCS character-set identifier

- Number of copies
- SYSOUT program name

The process symbol shows the procedure and procedure step name, or instream step name, along with the program name.

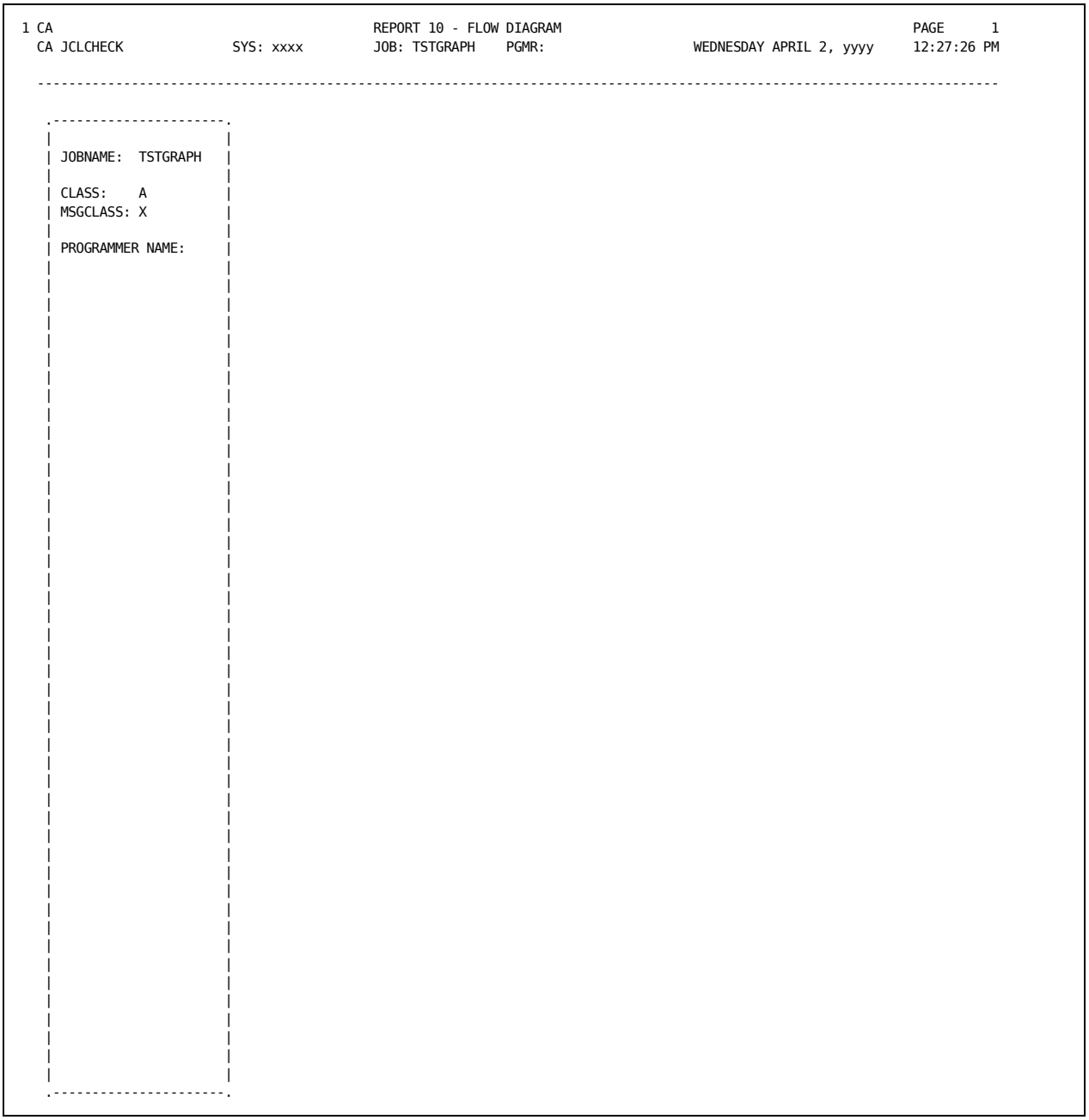
Standard flowchart graphic symbols are used to represent disk, tape, card and printed files, communications lines, and online terminals. For impact printers, the symbols themselves are labeled. Non-impact printers do not have labeled symbols, as they can achieve higher graphic resolution. Each symbol includes the ddname and the corresponding statement number from the JCL.

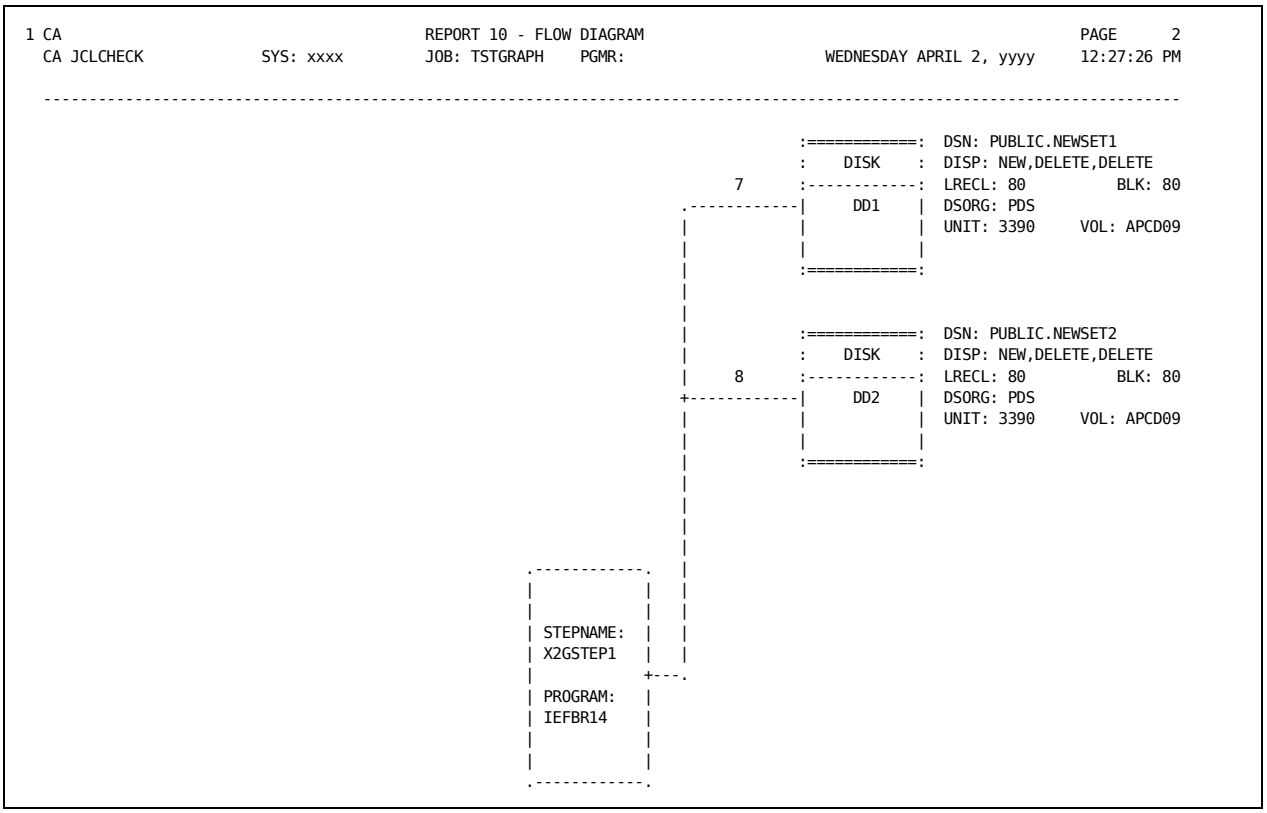
When a chart for one step extends beyond one page, the process symbol is repeated, with page connectors on each page.

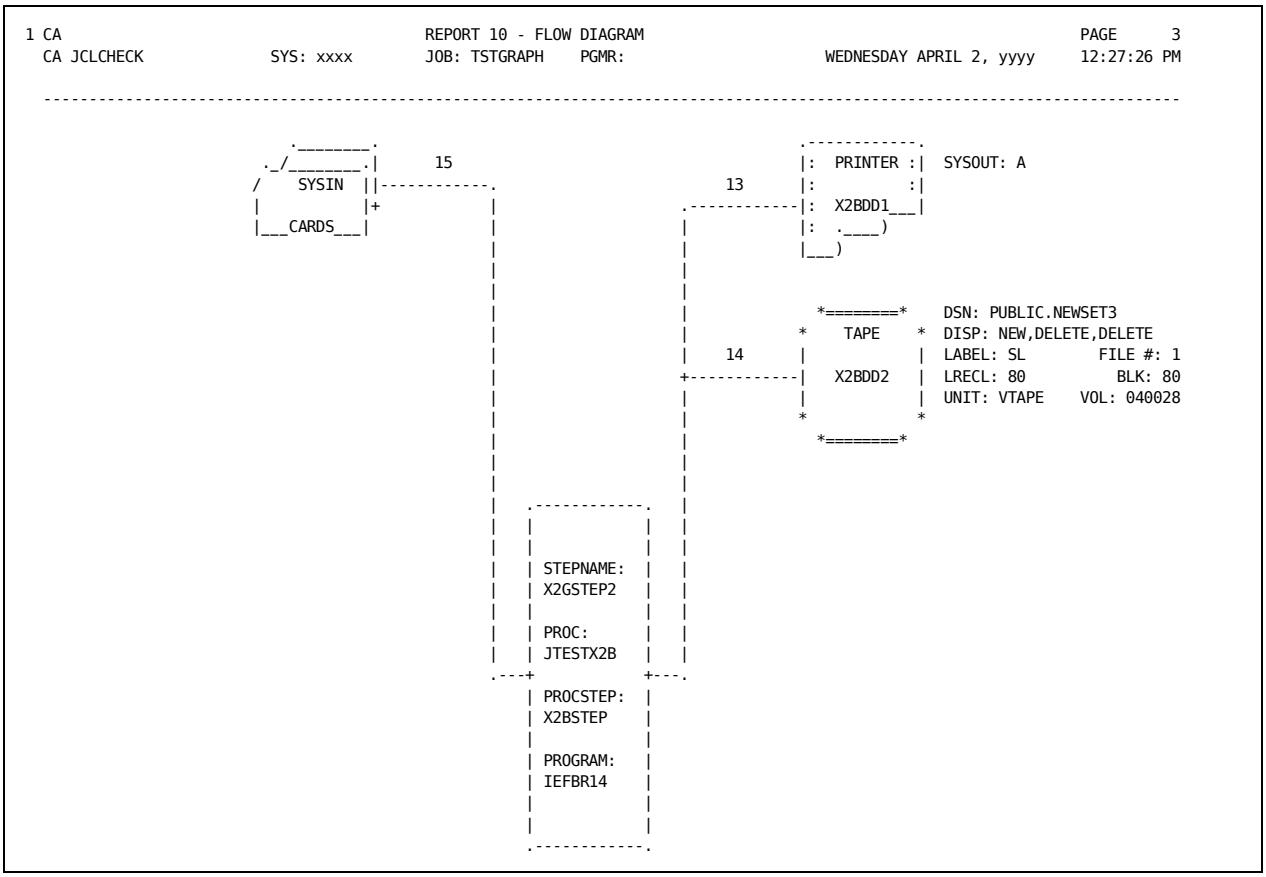
Concatenated DD statements are represented by having the file symbol duplicated. All concatenated files are listed in order of concatenation. Concatenated files that extend on to a second page do not generate repeated symbols.

Follow the instructions for Report 1 - LISTING OF JOBSTREAM JCL and specify the GRAPH option.

Sample Report 10 - Flow Diagram




1 CA REPORT 10 - FLOW DIAGRAM PAGE 2
CA JCLCHECK SYS: xxxx JOB: TSTGRAPH PGMR: WEDNESDAY APRIL 2, yyyy 12:27:26 PM
DISK
7
DD1
DSN: PUBLIC.NEWSET1
DISP: NEW,DELETE,DELETE
LRECL: 80 BLK: 80
DSORG: PDS
UNIT: 3390 VOL: APCD09
DISK
8
DD2
DSN: PUBLIC.NEWSET2
DISP: NEW,DELETE,DELETE
LRECL: 80 BLK: 80
DSORG: PDS
UNIT: 3390 VOL: APCD09
STEPNAME:
X2GSTEP1
PROGRAM:
IEFBR14



Report 11 - Procedure Cross-Reference

The Procedure Cross-Reference report lists all procedures used by a job and a cross-reference of all of the symbolic parameters used in each procedure. It is produced only if the PROCXREF option is selected.

Follow the instructions for Report 1 - LISTING OF JOBSTREAM JCL and specify the **PROCXREF** option.

Sample Report 11 - Procedure Cross-Reference

```
CA                                  REPORT 11 - PROCEDURE CROSS REFERENCE
CA JCLCHECK             SYS: xxxx     JOB: USER002         PGMR: JOB CONTROL COMPILE                THURSDAY JANUARY 9, yyyy
-------------------------------------------------------------------------------------------------------------------------------
PROCEDURE          LIBRARY   CONTAINING PROCEDURE                     REFERENCED IN
NAME               VOLUME    DSNAME                                  STATEMENT
-------------------------------------------------------------------------------------------------------------------------------

CAZ1ASM            APCM01    CAI.APCMTL.SPFLIB                       30
*ASM               'IEV90'                                           31  35
CAIMAC                       'CAI.APCMTL.CAIMAC'                     30  41
*CAIMAC                      'CAI.CAIMAC'                            31
M                            'CAZ1SUSR'                              30  44
*M                           'MEMBNAME'                              31
OPTLIB'                      'CAI.APCMTL.PPOPTION'                   30  44
*OPTLIB                      'CAI.PPOPTION'                          31
*SYSMAC                      'SYS1.MACLIB'                           31  42
CAZ1JCS            APCM01    CAI.APCMTL.SPFLIB                       13
CAILOAD                      'CAI.APCMTL.CAILOAD'                    13  26
OPTLIB                       'CAI.APCMTL.PPOPTION'                   13  28
RULES                        'CAI.APCMTL.JCLCHK.STDRULE'             13  27
CAZ1LNK            APCM01    CAI.APCMTL.SPFLIB                       46
CAILOAD                      'CAI.APCMTL.CAILOAD'                    46  52  53
*CAILOAD                              'CAI.CAILOAD'                          47
```

Audit Report

The JCL Processing Audit Report shows you a statistical summary, input records, and averages of jobs, steps, and DD statements.

Follow the instructions for Report 1 - LISTING OF JOBSTREAM JCL and use the **AUDIT** option.

Sample Audit report

```
                                    JCL PROCESSING AUDIT REPORT
CA JCLCHECK             SYS: xxxx                                                                                TUESDAY OCTOBER 24, yyyy
-------------------------------------------------------------------------------------------------------------------------------
                  -- (RECORDS) --                |          -- (TOTALS) --            |                -- (AVERAGES) --
         GRAND    SYSIN     PROCLIB   CTL-CARD   |                                    |     PROCS      STEPS      STEPS      DDS
         TOTAL    TOTAL     TOTAL     TOTAL     | JOBS     PROCS     STEPS     DDS |     PER JOB    PER PROC   PER JOB    PER STEP
-------------------------------------------------------------------------------------------------------------------------------

          29        6         23        0           1        3         8         7        3.0        2.7        8.0        .
#####################################################  End OF AUDIT REPORT FOR RUN  ##########################################
```

Note: For information about runtime options and selecting reports using the ISPF dialog, see the *Command Reference Guide*.

Defining Syntax Checking and Message Severity

CA JCLCheck performs complete syntax checking on the JCL including JES2 and JES3 control statements. CA JCLCheck reads, validates, and interprets certain IDCAMS, IEHPROGM, and XCOM control statements that perform JCL-like functions. CA JCLCheck simulates allocation/termination conditions to identify common errors such as misspelled data set names, incorrect disposition, and incorrect volume serial numbers.

Standard CA JCLCheck error messages are assigned to one of five value levels. These message levels are:

0 = Informational

4 = Warning

8 = Error

12 = Critical

16 = Severe

The CA JCLCheck SEV option allows you to define which level of error severity you want to view.

The instructions in this section show you how to use the CA JCLCheck SYNTAX option and SEV option to produce the following results:

- Viewing of CA JCLCheck error messages
- Elimination of warning messages

Viewing Error Messages

1. Create some errors in your JCL, such as **EXEC PPM=**.
2. Rerun CA JCLCheck.
3. Scan the CA JCLCheck reports to see the messages produced in Report 6 - Error Messages.

```
EDIT -- CAI.APCMTL.CHQAJCL(ZXJCSCMP) - 01.05 - MEMBER ZXJCSCMP SAVED
COMMAND ===>                                              SCROLL ===> PAGE

 //* PROCEDURE TO COMPILE JOB CONTROL STANDARDS FROM THE JCLCHECK
 //* JOB CONTROL STANDARDS (JCS) RULES DATABASE. THE OUTPUT OF THE
 //* COMPILE CONSISTS OF A SERIES OF MACROS THAT NEED TO BE ASSEMBLED
 //* AND LINKED AS MODULE 'CAZ1SUSR'.
 //* ------------------------------------------------------------------
 //EVAL1    EXEC PPM=CAZ1EVAL
 //STEPLIB  DD  DISP=SHR,                     LOADLIB FOR CAZ1EVAL
 //             DSN=CAI.APCMTL.CAILOAD
 //STDRULE  DD  DISP=SHR,                     JCLS RULES DATABASE
 //             DSN=CAI.APCMTL.JCLCHK.STDRULE
 //SYSLIN   DD  DISP=SHR,                     OUTPUT LIBRARY
 //             DSN=CAI.APCMTL.CAZ2OPTN(CAZ1SUSR)
 //SYSOUT  DD  SYSOUT=*
 //*-------------------------------------------------------------------*
 //*        PROCEDURE TO ASSEMBLE SOURCE FOR JCLCHECK USER EXITS       *
```

```
-----------------------CA JCLCHECK/INVOCATION -------------------------
COMMAND ===>

Option Selected: FOREGROUND
Date:  yyyy/mm/dd             User: USER02             Time: 11:35

Data Set Name         ===> 'CAI.APCMTL.CHQAJCL'
Library Type          ===> PDS      (SEQ, PDS, PAN, LIB)
Member                ===> ZXJCSCMP (Batch Submit -- Use * for all members)
Volume Seria l        ===>          (If Not CATALOGED)
CPU-ID                ===>          (Multi-CPU Only, blank: This CPU)
SYSTERM               ===> N        (Y/N/P, Error Statement and Messages only)
SYSPRINT              ===> Y        (Y/N/P, Reports 1-9 according to options )

Specify any additional options (FOR THIS RUN ONLY) below
OPTIONS               ===> CC(5)  CT    SP(RPT 65 NOALL) V
OPTIONS               ===> J F XREF PXR(RPT) SX RP ER( END)
OPTIONS               ===> SYN RUNT
OPTIONS               ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
OPTIONS               ===>
SAVED                         ===>
SAVED                         ===>
```

```
 BROWSE  -- USER02.JCLCHECK.SYSPRINT ------------------------------------------------ LINE 00000220 COL 001 080
 COMMAND ===>                                                                          SCROLL ===> PAGE

------------------------------------------------------------------------------------------------------------------
  STMT  MESSAGE
  NUM   NUMBER    MESSAGE TEXT
------------------------------------------------------------------------------------------------------------------

   1.   CAY6326S   'VSIGNON' UNKNOWN RETURN CODE 16 FROM CAISSF, SECURITY VALIDAT
        CAY6323I   FEEDBACK= 'CAS9TSS: UNKNOWN FUNCTION NOT SUPPORTED'
  25.   CAY6025E   REQUIRED PGM OR PROC PARAMETER MISSING
        CAY6037E   UNKNOWN KEYWORD 'PPM' SPECIFIED IN OPERAND OF 'EXEC'
  27.   CAY6077W   NO '3050200E' UNITS ARE AVAILABLE FOR 'ARCHIV'
        CAY6067W   UNABLE TO VERIFY THAT DATA SET 'CAI.APCMTL.JCLCHK.STDRULE
  33.   CAY6051E   STATEMENT REFERENCED BY BACK REFERENCE 'EVAL1' CANNOT BE FOUND
        CAY6145I   CONDITION CODE PROCESSING TERMINATED DUE TO ERRORS
  38.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S0
CAY6000 5 STATEMENTS FLAGGED IN JOB 'USER02' MAXIMUM SEVERITY WAS 12
```

Eliminating Warning Messages

1. Specify **SEV(8)** in the OPTIONS field on the INVOCATION panel.

   Specifying SEV(8) causes CA JCLCheck to display only those errors at a severity level of 8 or higher. These errors discontinue JCL execution.

2. Rescan the JCL and view Report 6 again. Notice that SEV(8) has suppressed the Warning messages.

```
---------------------- CA JCLCHECK/INVOCATION -------------------------
COMMAND ===>

Option Selected: FOREGROUND

Date:  yyyy/mm/dd             User: USER02             Time: 11:35

Data Set Name  ===> 'CAI.APCMTL.CHQAJCL'
Library Type   ===> PDS                (SEQ, PDS, PAN, LIB)
Member         ===> ZXJCSCMP  (Batch Submit -- Use * for all members)
Volume Serial  ===>           (If Not CATALOGED)
CPU-ID         ===>           (Multi-CPU Only, blank: This CPU)
SYSTERM        ===> N         (Y/N/P, Error Statement and Messages only)
SYSPRINT       ===> Y         (Y/N/P, Reports 1-9 according to options )

Specify any additional options (FOR THIS RUN ONLY) below
OPTIONS        ===> CC(5)  CT    SP(RPT 65 NOALL) V
OPTIONS        ===> J F XREF PXR(RPT) SX RP ER( END)
OPTIONS        ===> SYN RUNT SEV(8)
OPTIONS        ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
OPTIONS        ===>
SAVED                  ===>
SAVED                  ===>
```

```
 BROWSE -- USER02.JCLCHECK.SYSPRINT ------------------------------------------------------------- LINE 00000220 COL 001 080
 COMMAND ===>                                                                                      SCROLL ===> PAGE

-------------------------------------------------------------------------------------------------------------------------------
  STMT  MESSAGE
  NUM   NUMBER    MESSAGE TEXT
-------------------------------------------------------------------------------------------------------------------------------

   1.   CAY6326S   'VSIGNON' UNKNOWN RETURN CODE 16 FROM CAISSF, SECURITY VALIDAT
  25.   CAY6025E   REQUIRED PGM OR PROC PARAMETER MISSING
        CAY6037E   UNKNOWN KEYWORD 'PPM' SPECIFIED IN OPERAND OF 'EXEC'
  33.   CAY6051E   STATEMENT REFERENCED BY BACK REFERENCE 'EVAL1' CANNOT BE FOUND
CAY6000 3 STATEMENTS FLAGGED IN JOB 'USER02' MAXIMUM SEVERITY WAS 12
```

Note: For information about the runtime option SEV, see the *Command Reference Guide*. For information about CA JCLCheck messages, see the *Message Reference Guide*.

Defining Runtime

CA JCLCheck checks for execution-time errors, which result in a system ABEND. The following JCL problems can cause execution-time errors: missing or invalid programs, missing data sets, and incorrect DCB information. CA JCLCheck makes special checks for other common errors such as, incorrect order of cataloged procedure overrides. You can also see flagged runtime errors from the syntax checking.

This section shows you how to:

- Simulate Runtime Errors (see page 46)
- Perform Runtime Checking (see page 46)
- Deactivate Runtime Checking (see page 47)

Simulate Runtime Errors

To simulate runtime errors, modify your JCL to reference a data set that does not exist, as shown below.

```
EDIT ------ CAI.APCMTL.CHQAJCL(USER002) - 01.02 --------- COLUM NS 001 072
COMMAND ===>                                               SCROLL ===> PAGE

  000100 //USER002 JOB '40700000',JOEL,CLASS=T,MSGCLASS=X,NOTIFY=RAVJO02,
  000200 //   REGION=4096K
  000700 //DSDUMP1 EXEC PGM=ADRDSSU
  000800 //SYSPRINT  DD SYSOUT=*
  000900 //TAPE1     DD DSN=USER001.R60JCK.BKUP(+0),DISP=OLD,
  001000 //             UNIT=(CART,,DEFER)
  001100 //SYSIN     DD *
  001200   REST IDD(TAPE1) -
  001400   ODY((WORK81,3380)  -
  001410       (WORK82,3380)) -
  001500   DS(INC(USER002.**)) -
  001610   LVOL(WORK83,WORK84) -
  001620   CAT
  001700 /*
```

Perform Runtime Checking

1. Specify the runtime option **RUNT** in the OPTIONS field on the CA JCLCheck INVOCATION panel.
2. Rerun CA JCLCheck to rescan the JCL. Note REPORT 6 - ERROR MESSAGES.

```
 --------------------------- CA JCLCHECK INVOCATION ---------------------
  COMMAND ===>


Option Selected: FOREGROUND

Date:  yyyy/mm/dd              User: USER02             Time: 14:26

 Data Set Name===> 'CAI.APCMTL.CHQAJCL'
 Library Type ===> PDS         (SEQ, PDS, PAN, LIB)
 Member       ===> USER02      (Batch Submit -- Use * for all members)
 Volume Serial===>             (If Not CATALOGED)
 CPU-ID       ===>             (Multi-CPU Only, blank: This CPU)
 SYSTERM      ===> N           (Y/N/P, Error Statement and Messages only)
 SYSPRINT     ===> Y           (Y/N/P, Reports 1-9 according to options )


 Specify any additional options (FOR THIS RUN ONLY) below
 OPTIONS      ===> CC(5)  CT    SP(RPT 65 NOALL) V PROCX
 OPTIONS      ===> J F XREF PXR(RPT) SX RP ER( END)
 OPTIONS      ===> SYN RUNT
 OPTIONS      ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
 OPTIONS      ===>
 SAVED                ===>
 SAVED                ===>
```

```
BROWSE -- USER02.JCLCHECK.SYSPRINT ------------------------------------------------------------ LINE 00000000 COL 001 080
COMMAND ===>                                                                                    SCROLL ===> PAGE

 JCLCHECK         INVOKED AT  2:26:33 PM ON FRIDAY NOVEMBER 12, yyyy
 SYSTEM  INFORMATION: CPU MODEL=9021 SP4.3.0 TSO2.04.0 SMS1110 ENVIRONMENT ARRAY: 100000000000000000 000
 CALLING  PARAMETERS: O(OPTS)
 DEFAULT  PARAMETERS: ACT,SYNTAX,RUNTIME
 OPTS     PARAMETERS: CC(5)  CT    SP(RPT 65 NOALL) V PROCX
                      J F XREF PXR(RPT) SX RP ER( END)
                      SYN RUNT
                      SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
                      PDS INCLUDE(USER02)
 OPTIONS IN EFFECT: ACTION CCLIST(5) CTLSCAN ERROR(END) FULLLIST OPTIONS(OPTS)
                      PDS($ 9999999999) PROCXREF PULL PXREF(RPT) RP
                      SPACE(REPORT 65 NOALL) SXREF(RPT) VSAM() XREF
 SECURITY OPTIONS IN EFFECT: DATASET DASDVOL MGMTCLAS STORCLAS PROGRAM(DEFINE) NO


 CA                                    REPORT 1 - LISTING OF JOBSTREAM JCL
 CA JCLCHECK              SYS: xxxx   JOB: USER002         PGMR: CA01                      THURSDAY JANUARY 9, yyyy
 ------------------------------------------------------------------------------------------------------------------------------
 STMT.
 NUM  STATEMENT TEXT
 ------------------------------------------------------------------------------------------------------------------------------
      ……..…+….……1……....+….…...2….……+…….……3.……..+…….……4….……+…….....5……....+…….....6.…...+.…...7
  1.  //USER02 JOB '40700000',JOEL,CLASS=T,MSGCLASS=X,NOTIFY=USER02,
      //  REGION=4096K
```

```
CA                                    REPORT 6 - ERROR MESSAGES
CA JCLCHECK              SYS: xxxx     JOB: USER02          PGMR: CA01                              THURSDAY JANUARY 9, yyyy
------------------------------------------------------------------------------------------------------------------------------
  STMT  MESSAGE
  NUM   NUMBER     MESSAGE TEXT
------------------------------------------------------------------------------------------------------------------------------

   1.   CAY6326S   'VSIGNON' UNKNOWN RETURN CODE 16 FROM CAISSF, SECURITY VALIDATATION
        CAY6323I   FEEDBACK= 'CAS9TSS: UNKNOWN FUNCTION NOT SUPPORTED'
   9.   CAY6186E   GDG INDEX NOT BUILT
        CAY6079E   DATA SET 'USER02.R60JCK.BKUP(0)' SPECIFIED AS OLD OR SHR, BUT CANNOT BE FOUND
   CAY6000 2 STATEMENTS FLAGGED IN JOB 'USER02' MAXIMUM SEVERITY WAS 12
```

Deactivate Runtime Checking

You may want to deactivate runtime checking if, for instance, you have data sets and programs within your JCL that do not yet exist. In this case, you would not want to use runtime checking because CA JCLCheck would produce errors for those items that do not exist. Turning off runtime checking speeds up processing since it eliminates data set and execution step existence checking. To deactivate runtime checking:

1. Specify the NORUNTIME option **NORUNT** in the OPTIONS field on the CA JCLCheck INVOCATION panel.
2. Rerun CA JCLCheck to rescan the JCL. Note the change in REPORT 6 - ERROR MESSAGES.

```
------------------ CA JCLCHECK/INVOCATION ----------------------------
COMMAND ===>

Option Selected: FOREGROUND

Date:  yyyy/mm/dd             User: USER02             Time: 14:26

Data Set Name ===> 'CAI.APCMTL.CHQAJCL'
Library Type  ===> PDS                 (SEQ, PDS, PAN, LIB)
Member        ===> USER02     (Batch Submit -- Use * for all members)
Volume Serial ===>            (If Not CATALOGED)
CPU-ID        ===>            (Multi-CPU Only, blank: This CPU)
SYSTERM       ===> N          (Y/N/P, Error Statement and Messages only)
SYSPRINT      ===> Y          (Y/N/P, Reports 1-9 according to options )


Specify any additional options (FOR THIS RUN ONLY) below
OPTIONS       ===> CC(5)  CT    SP(RPT 65 NOALL) V PROCX
OPTIONS       ===> J F XREF PXR(RPT) SX RP ER( END)
OPTIONS       ===> SYN NORUNT
OPTIONS       ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
OPTIONS       ===>
SAVED                ===>
SAVED                ===>
```

```
   BROWSE -- USER02.JCLCHECK.SYSPRINT ------------------------------------------------------ LINE 00000000 COL 001 080
   COMMAND ===>                                                                                 SCROLL ===> PAGE

JCLCHECK        INVOKED AT  2:27:27 PM ON FRIDAY NOVEMBER 12, yyyy
SYSTEM  INFORMATION: CPU MODEL=9021 SP4.3.0 TSO2.04.0 SMS1110 ENVIRONMENT ARRAY: 10000000000000000000
CALLING  PARAMETERS: O(OPTS)
DEFAULT  PARAMETERS: ACT,SYNTAX,RUNTIME
OPTS     PARAMETERS: CC(5)  CT    SP(RPT 65 NOALL) V PROCX
                     J F XREF PXR(RPT) SX RP ER( END)
                     SYN NORUNT
                     SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
                     PDS INCLUDE(USER02)
OPTIONS IN EFFECT: ACTION CCLIST(5) CTLSCAN ERROR(END) FULLLIST OPTIONS(OPTS)
                     PDS($ 9999999999) PROCXREF PULL PXREF(RPT) RP
                     SPACE(REPORT 65 NOALL) SXREF(RPT) JESCHEK XREF NOR
SECURITY OPTIONS IN EFFECT: DATASET DASDVOL MGMTCLAS STORCLAS PROGRAM(DEFINE) NO

CA                                  REPORT 1 - LISTING OF JOBSTREAM JCL
CA JCLCHECK              SYS: xxxx     JOB: USER002         PGMR: CA01                          THURSDAY JANUARY 9, yyyy
-------------------------------------------------------------------------------------------------------------------------
 STMT
 NUM   STATEMENT TEXT
-------------------------------------------------------------------------------------------------------------------------
       .…+….....1……....+….…...2……....+….....3.……..+……....4……....+….…..5….…...+……....6….….…+.….....7

  1. //USER02 JOB '40700000',JOEL,CLASS=T,MSGCLASS=X,NOTIFY=USER02,
     //  REGION=4096K
```

```
CA                                  REPORT 6 - ERROR MESSAGES
CA JCLCHECK              SYS: xxxx     JOB: USER02          PGMR: CA01                          THURSDAY JANUARY 9, yyyy
-------------------------------------------------------------------------------------------------------------------------
STMT   MESSAGE
NUM    NUMBER    MESSAGE TEXT
-------------------------------------------------------------------------------------------------------------------------
CAY6000  NO STATEMENTS FLAGGED IN JOB 'USER02' MAXIMUM SEVERITY WAS 0
```

Note: For information about runtime options, see the *Command Reference Guide*. For information about CA JCLCheck messages, see the *Message Reference Guide*.

Invoking CA JCLCheck with EDCHEK

The EDCHEK option of CA JCLCheck runs as an edit macro under ISPF to allow you to validate your JCL from within ISPF edit. You can analyze and report on the JCL you are currently editing.

To try EDCHEK, execute CLIST CAZ1SEDC in CAZ2CLS0 to allocate the files you need then select ISPF to edit a JCL member. While trying EDCHEK, you may want to modify your JCL to generate errors.

This feature offers the ideal functionality for checking your JCL online while working in the ISPF Editor. You can set up a PFKEY for the EDCHEK command to invoke CA JCLCheck.

Note: For information about the installation of the EDCHEK feature, see the *Installation Guide*. For information about EDCHEK control statements, see the *Programming Guide*. For information about the EDCHEK runtime option, see the *Command Reference Guide*.

You can use either the %EJCK or EJCK command to invoke EDCHEK; however, we recommend that you use %EJCK to reduce processing time.

To get acquainted with the EDCHEK feature of CA JCLCheck, use the instructions on the following pages to do the following tasks:

- Invoke EDCHEK as an Edit Macro (see page 50)
- Change Runtime Options Prior to EDCHEK Invocation (see page 51)
- Specify Options (see page 52)
- View Reports (see page 53)
- Continue Editing JCL (see page 54)

Invoke EDCHEK as an Edit Macro

To invoke EDCHEK as an edit macro:

1. From the TSO/ISPF Main menu, select the ISPF Edit function and access a JCL member that you want to check.
2. Enter **%EJCK** on the command line.

```
 EDIT ---- CAI.CHQA.JCLCHECK.TESTJCL(JTEST11) - 01.15 ------ COLUMNS 001 072
 COMMAND ===> %EJCK                                        SCROLL ===> PAGE

000001 //JTEST111 JOB '40100000',JTEST111,CLASS=K,MSGCLASS=X,
000002 //  RESTART=(JS020.PS010)
000003 //*
000004 //* THIS TESTJOB TESTS THE VALIDITY OF VARIOUS REFERBACKS IN JCL
000005 //*
000006 //TESTPROC PROC
000007 //PS010  EXEC PGM=IEFBR14
000008 //DD011  DD  DISP=SHR,
000009 //           DSN=CAI.CHQA.TESTJCL,
000010 //           UNIT=SYSDA,
000011 //           VOL=SER=APCM04
000012 //DD012  DD  DUMMY
000013 //TESTPROC PEND
000014 //OUT1 OUTPUT CLASS=X
```

The following is a sample of the top of a screen containing EDCHEK output.

```
EDIT ---- CAI.CHQA.JCLCHECK.TESTJCL(JTEST11) - 01.15 ------ COLUMNS 001 072
COMMAND ===>                                              SCROLL ===> PAGE

==MSG> CAY6000 9 STATEMENTS FLAGGED IN JOB "JTEST111" MAXIMUM SEVERITY WAS 8
==MSG>
==MSG>
==MSG>
000001 //JTEST111 JOB '40100000',JTEST111,CLASS=K,MSGCLASS=X,
000002 //  RESTART=(JS020.PS010)
000003 //*
000004 //* THIS TESTJOB TESTS THE VALIDITY OF VARIOUS REFERBACKS IN JCL
```

The following shows samples of EDCHEK messages further down in the JCL.

```
000087 //      VOL=REF=*.DD041
==MSG> //      VOL=REF=*.DD041
==MSG> CAY6051E STATEMENT REFERENCED BY BACK REFERENCE "*.DD041" CANNOT BE
==MSG>      FOUND. (USED IN "REF" PARAMETER OF "VOL")
==MSG>
000088 //JS060  EXEC PGM=IEFBR14,
==MSG> //JS060  EXEC PGM=IEFBR14,
000089 //       COND=(0,LT,JS070)
==MSG> //       COND=(0,LT,JS070)
==MSG> CAY6051E STATEMENT REFERENCED BY BACK REFERENCE "JS070" CANNOT BE
==MSG>          FOUND. (USED IN "STEP" PARAMETER OF "COND#1")
==MSG> CAY6145I CONDITION CODE PROCESSING TERMINATED DUE TO ERRORS
==MSG>
000090 //*
```

Change Runtime Options Prior to EDCHEK Invocation

1. On the command line, enter **%EJCK PANEL** to get the options panel.
2. On the JCK OPTIONS panel, specify the options you want to use and press **F3** (End) to exit the panel.

   This automatically invokes EDCHEK using the options you specified.

```
EDIT ---- CAI.CHQA.JCLCHECK.TESTJCL(JTEST11) - 01.15 ------ COLUMNS 001 072
COMMAND ===> %EJCK PANEL                                  SCROLL = ==> PAGE

000001 //JTEST111 JOB '40100000',JTEST111,CLASS=K,MSGCLASS=X,
000002 //  RESTART=(JS020.PS010)
000003 //*
000004 //* THIS TESTJOB TESTS THE VALIDITY OF VARIOUS REFERBACKS IN JCL
000005 //*
000006 //TESTPROC PROC
000007 //PS010  EXEC PGM=IEFBR14
000008 //DD011  DD  DISP=SHR,
000009 //           DSN=CAI.CHQA.TESTJCL,
000010 //           UNIT=SYSDA,
000011 //           VOL=SER=APCM04
000012 //DD012  DD  DUMMY
000013 //TESTPROC PEND
000014 //OUT1 OUTPUT CLASS=X
```

```
JCK21 --------------- CA JCLCheck xx.x OPTIONS -----------------------------
COMMAND ===>
 DATE: yyyy/mm/dd              USER: USERID1                      TIME: hh:mm

STANDARD - Name of STANDARD for compliance checking              =>
             =>           =>           =>           =>           =>
             =>           =>
STDREXX  - Name of REXX EXEC for compliance checking             =>
SYNTAX   - Do syntax checking of JCL                             => Y
RUNTIME  - Do runtime checking of JCL (all runtime checks)       => Y
REMOTE   - Do remote validation of JCL                           => N
      =>     =>
RESOLVE  - Resolve external variables by calling product. . . .  => CA7
   => E   DSN=> USERID1.JCK.CNTL                                MEM=> RESPARMS
EDCHKLB  - Generate ISPF labels for lines with errors              => N
EDCHKSL  - Display summary of errors                               => Y
SECURITY - Do Security Checking                                    => N
USER     - Specify Userid for security checking                    =>

    BYPASS THIS SCREEN NEXT TIME.............................. => N
```

Specify Options

To specify options that do not exist on the JCK OPTIONS panel or to override the options specified on the JCK OPTIONS panel, enter **%EJCK** on the COMMAND line along with additional options such as **MVSL(4) and SEV(8)**.

```
EDIT ---- CAI.CHQA.JCLCHECK.TESTJCL(JTEST11) - 01.15 ----- COLUMNS 001 072
COMMAND ===> %EJCK MVSL(4) SEV(8)                        SCROLL ==> PAGE

 000001 //JTEST111 JOB '40100000',JTEST111,CLASS=K,MSGCLASS=X,
 000002 //  RESTART=(JS020.PS010)
 000003 //*
 000004 //* THIS TESTJOB TESTS THE VALIDITY OF VARIOUS REFERBACKS IN JCL
 000005 //*
 000006 //TESTPROC PROC
 000007 //PS010  EXEC PGM=IEFBR14
 000008 //DD011  DD  DISP=SHR,
 000009 //           DSN=CAI.CHQA.TESTJCL,
 000010 //           UNIT=SYSDA,
 000011 //           VOL=SER=APCM04
 000012 //DD012  DD  DUMMY
 000013 //TESTPROC PEND
 000014 //OUT1 OUTPUT CLASS=X
```

View Reports

1. On the command line, enter **%EJCK REPORTS**, and press **F3** (End).

   Note: Entering **Y** or **N** in the Bypass This Screen Next Time field on the JCK OPTIONS panel determines whether or not you access the JCK OPTIONS panel before you invoke EDCHEK.

2. Specifying REPORTS produces CA JCLCheck reports.

3. When you press **F3** (End) to end out of the reports, EDCHEK returns you to the EDCHEK panel with the error messages displayed.

```
EDIT ---- CAI.CHQA.JCLCheck.TESTJCL(JTEST11) - 01.15 ------ COLUMNS 001 072
COMMAND ===> %EJCK REPORTS                                  SCROLL ===> PAGE

 000001 //JTEST111 JOB '40100000',JTEST111,CLASS=K,MSGCLASS=X,
 000002 //  RESTART=(JS020.PS010)
 000003 //*
 000004 //* THIS TESTJOB TESTS THE VALIDITY OF VARIOUS REFERBACKS IN JCL
 000005 //*
 000006 //TESTPROC PROC
 000007 //PS010  EXEC PGM=IEFBR14
 000008 //DD011  DD  DISP=SHR,
 000009 //           DSN=CAI.CHQA.TESTJCL,
 000010 //           UNIT=SYSDA,
 000011 //           VOL=SER=APCM04
 000012 //DD012  DD  DUMMY
 000013 //TESTPROC PEND
 000014 //OUT1 OUTPUT CLASS=X
```

This is a sample of some of the report panels:

```
BROWSE -- USER02.EDCHEK.SYSPRINT ------------- LINE 0000 LAST CC WAS  08
COMMAND ===>                                              SCROLL ===>

CA JCLCHECK          INVOKED AT 11:06:29 AM ON WEDNESDAY NOVEMBER 10, yyyy
SYSTEM  INFORMATION: CPU MODEL=9021 SP4.3.0 TSO2.04.0 SMS1110 ENVIRONMENT
ARRAY:
CALLING  PARAMETERS:  T OPT(EDCHKDD )  RUNT SYN NOSEC USER(USER02) NOMV S
DEFAULT  PARAMETERS:  ACT,SYNTAX,RUNTIME
EDCHKDD  PARAMETERS:    EDCHEK XREF PXREF RPT SXREF -
                                EASY NOCOND SP(RPT 65) ERR(I E) -
                                DEST(J) NOSP -
                                LIST(EDCPRINT) -
                                PROC(PROC00)
                        ALLOCATE FI(PROC00) DA(SYS1.PROCLIB)
                        ALLOC FI(*) DA(SYS2.PROCLIB)
                        ALLOC FI(*) DA(USER.PROCLIB)
                        GO
OPTIONS IN EFFECT: ACTION CTLSCAN DESTCHK(JES) EASYPROC ERROR(END) FULLLIST
                   NOSPIE OPTIONS(EDCHKDD) PROC(PROC00) PULL PXREF(RPT)
                   RPTRPT RUNTIME SPACE(REPORT 65) SXREF(RPT)
                   TERM(OPROC)TXREF NOCOND
```

```
CA                                  REPORT 6 - ERROR MESSAGES
CA JCLCHECK            SYS: xxxx       JOB: JTEST111        PGMR: JTEST111                          THURSDAY JANUARY 9, yyyy
-------------------------------------------------------------------------------------------------------------------------------
STMT     MESSAGE
NUM      NUMBER     MESSAGE TEXT
-------------------------------------------------------------------------------------------------------------------------------
 21.     CAY6148W   DSNAME 'CAI.CHQA.J661.DD021' FALLS BELOW MINIMUM SPECIFIED VAL
 26.     CAY6148W   DSNAME 'CAI.CHQA.J661.DD031' FALLS BELOW MINIMUM SPECIFIED VAL
 36.     CAY6352E   REFERBACK STATEMENT CANNOT FOLLOW A FORWARD REFERENCE TO THIS
 46      CAY6051E   STATEMENT REFERENCED BY BACK REFERENCE '*.DD054' CANNOT BE FOU
 49.     CAY6051E   STATEMENT REFERENCED BY BACK REFERENCE '*.DD041' CANNOT BE FOU
 50.     CAY6051E   STATEMENT REFERENCED BY BACK REFERENCE 'JS070' CANNOT BE FOUND
         CAY6145I   CONDITION CODE PROCESSING TERMINATED DUE TO ERRORS
 55.     CAY6114W   STATEMENT 9 REFERENCED BY BACK REFERENCE 'JS010.PS010' IS IN A
 60.     CAY6051E   STATEMENT REFERENCED BY BACK REFERENCE 'JS020' CANNOT BE FOUND
 76.     CAY6076E   UNIT 'SYSDA' INCONSISTENT WITH VOLUME 'ABCDEF'
         CAY6067W   UNABLE TO VERIFY THAT DATA SET 'CAI.CHQA.DD102' IS ON VOLUME '
         CAY6148W   DSNAME 'CAI.CHQA.DD102' FALLS BELOW MINIMUM SPECIFIED VALUE OF
         CAY6086E   BLKSIZE INCONSISTENT WITH LRECL FOR DATA SET WITH FIXED RECORD
```

```
EDIT ----- CAI.CHQA.JCLCheck.TESTJCL(JTEST11) - 01.15 ----- COLUMNS 001 072
COMMAND ===>                                               SCROLL ===> PAGE

==MSG> CAY6000 11 STATEMENTS FLAGGED IN JOB "JTEST111" MAXIMUM SEVERITY WAS 8
==MSG>
==MSG>
==MSG>
000001 //JTEST111 JOB '40100000',JTEST111,CLASS=K,MSGCLASS=X,
000002 //  RESTART=(JS020.PS010)
000003 //*
000004 //* THIS TESTJOB TESTS THE VALIDITY OF VARIOUS REFERBACKS IN JCL
```

Continue Editing JCL

To continue editing JCL, enter the **RESET** command on the command line.

```
EDIT ----- CAI.CHQA.JCLCheck.TESTJCL(JTEST11) - 01.15 ------- COLUMNS 001 072
 COMMAND ===>  RESET                                         SCROLL ===> PAGE

 ==MSG> CAY6000 11 STATEMENTS FLAGGED IN JOB "JTEST111" MAXIMUM SEVERITY WAS 8
 ==MSG>
 ==MSG>
 ==MSG>
 000001 //JTEST111 JOB '40100000',JTEST111,CLASS=K,MSGCLASS=X,
 000002 //  RESTART=(JS020.PS010)
 000003 //*
 000004 //* THIS TESTJOB TESTS THE VALIDITY OF VARIOUS REFERBACKS IN JCL
```

Invoking CA JCLCheck with SUBCHEK

CA JCLCheck front-ends the TSO submit command through its SUBCHEK feature. This feature allows CA JCLCheck to analyze the JCL being submitted before the JCL is sent to the internal reader. If any errors are identified, the CA JCLCheck messages are displayed at your terminal, and consequently the JCL is *not* submitted for processing.

There are two ways you can control SUBCHEK:

- Specify runtime options and PROCLIBs to validate the JCL prior to submission.
- Specify whether SUBCHEK only validates the JCL or also passes it directly to SUBMIT without validation.

Runtime options and PROCLIBs are specified in a control file allocated to SUBCHKDD. Examples of this file, CAZ1SSUP and CAZ1SSUA, are provided in CAZ2OPTN. CAZ1SSUP shows you how to define the PROCLIB allocations and CAZ1SSUA is an example of how to set up SUBCHEK using AUTOPROC, which you do not need PROCLIB allocations.

If the control file SUBCHKDD is not allocated, SUBCHEK does not validate the JCL, but passes it directly to submit. You can use a sample CLIST, SUBCHK, in CAZ2CLS0 to allocate and de-allocate SUBCHKDD at your discretion. SUBCHK OFF de-allocates SUBCHKDD prior to invoking SUBCHEK and the JCL is not validated. SUBCHK ON allocates SUBCHKDD, causing validation of the JCL prior to submission. Tailor this CLIST to allocate the appropriate file and to allocate and de-allocate the CA TLMS CAIVMFI file if you are using the TLMS option.

You can use the CAZ1JSDX exit to dynamically allocate SUBCHKDD if it is not allocated to the TSO session. This is useful for sites that want to force certain users to use SUBCHEK. See the comments contained in the source for the CAZ1JSDX module on CAZ2OPTN.

You can allocate the control file in your LOGON procedures. You can also allocate or free the control file by command or by a CLIST. A sample CLIST has been provided (SUBCHK in CAZ2CLS0), which enables you to turn SUBCHEK on or off.

Note: For information about installing the SUBCHEK feature, see the *Installation Guide*. For information about SUBCHEK control statements, see the *Programming Guide*. For information about the SUBCHEK runtime option, see the *Command Reference Guide*.

Setting-Up Security

CA JCLCheck provides an interface to CA ACF2, CA Top Secret, and the IBM RACF security facilities. The CA JCLCheck pre-validation of the security environment helps you to reduce the incidence of S913 ABENDS and job failures because of insufficient access authority to DATA SET, DASDVOL, PROGRAM, STORCLAS, and MGMTCLAS resources.

CA JCLCheck performs security checks that are based on the user ID identified in either the USER parameter of the JOB statement or the CA ACF2 //*LOGONID statement. If a user ID does not exist in either of these places, then by default, the user ID of the person submitting the JCL is used. CA JCLCheck can also determine security checking when you specify a user ID with the USER option.

You can specify the USER option when you want to simulate security checking for the user ID of someone else. For example, you may want to set up security checking of another user if you are testing JCL to run in production under the user ID of someone else.

Note: For more information about security, see the *Installation Guide* and the *Programming Guide*.

Use the instructions in the following sections to perform the following tasks:

- Check a User ID for Security (see page 57)
- Create an Error Situation to Test Security (see page 59)

Check a User ID for Security

1. Enter **0** on the CA JCLCheck/SPF Menu, press enter.
2. Enter **3** on the CA JCLCheck Control Options Menu.

```
JCK0200-------------- CA JCLCHECK/SPF MENU ------------------------
OPTION ===> 0


  Date:  yyyy/mm/dd     User: USERID1    Level: xx.x

       0  -  Option Specifications

       1  -  CA JCLCHECK - Foreground Invocation

       2  -  CA JCLCHECK - Foreground Invocation / Submit

       3  -  CA JCLCHECK - Batch Submit

       4  -  CA 7 Interface Invocation

       5  -  CA Scheduler Interface Invocation

       6  -  JCLNeat - Batch Submit

       7  -  RESOLVE - External Variable Resolution

       X  -  Exit
```

```
 ----------------- CA JCLCHECK/CONTROL OPTIONS MENU -------------------
OPTION ===> 3


Date: yyyy/mm/dd      User: USER02             Time: 10:58

     1 - Process Control Options

     2 - Output Control Options

     3 - Security Options

     4 - JCLNeat Options

     5 - Job Card Specifications

     X - Exit
```

1. Do one of the following:
   - Enter a **user ID** in the **USER** field in the Security Options panel, and press **F3** (End).
   - Enter **USER(user ID)** in the **OPTIONS** field on one of the INVOCATION panels.

2. Do one of the following:

   - Enter Y in the SECURITY field in the Security Options panel, and press F3 (End) to exit the panel.
   - Enter **SEC(options)** (options being PROG, DASD, STORclas, or MGMTclas) in the OPTIONS field on one of the INVOCATION panels.

```
 ----------------- CA JCLCHECK SECURITY OPTIONS  ----------------------
COMMAND ===>

Date: mm/dd/yy.jjj User: USER02              Time:  16:27

SECURITY  - Do Security Checking                                  => Y
USER      Alternate user ID for security checking or blank        => USER02
DASDVOL   - Perform DASDVOL security checks                       => Y
MGMTCLAS  - Perform MGMTCLAS security checks                      => Y
STORCLAS  - Perform STORCLAS security checks                      => Y
PROGRAM   - Perform PROGRAM level security checks                 => Y
          - Check if executed program is defined to security    => Y
RUNCPU    - Alternate CPU ID for security validation or blank   =>
RUNDATE   - Future run date and time for security validation
          - Specify date in mm/dd/yy format or blank            =>
          - Specify time in hh:mm format or blank               =>


                  Update Profile With These Options             => Y
```

```
--------------------- CA JCLCHECK/INVOCATION --------------------------
COMMAND ===>

Option Selected: FOREGROUND

Date: yyyy/mm/dd          User: USER02              Time: 10:59

Data Set Name ===> 'CAI.APCMTL.CHQAJCL'
Library Type ===> PDS      (SEQ, PDS, PAN, LIB)
Member     ===> USER02     (Batch Submit -- Use * for all members)
Volume Serial ===>         (If Not CATALOGED)
CPU-ID     ===>            (Multi-CPU Only, blank: This CPU)
SYSTERM    ===> N          (Y/N/P, Error Statement and Messages only)
SYSPRINT   ===> Y          (Y/N/P, Reports 1-9 according to options )

Specify any additional options (FOR THIS RUN ONLY) below
OPTIONS    ===> CC(5)  CT    SP(RPT 65 NOALL) V
OPTIONS    ===> J F XREF PXR(RPT) SX RP ER( END)
OPTIONS    ===> SYN RUNT JOB USER(USER02)
OPTIONS    ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
OPTIONS    ===>
SAVED      ===>
SAVED      ===>
```

Create an Error Situation to Test Security

1. Change the USER or the JCL to contain errors.
2. Use one of the INVOCATION panels to run CA JCLCheck against a JCL member and view the results.

```
Specify any additional options (FOR THIS RUN ONLY) below
OPTIONS       ===> CC(5)  CT    SP(RPT 65 NOALL) V
OPTIONS       ===> J F XREF PXR(RPT) SX RP ER( END)
OPTIONS       ===> SYN RUNT JOB USER(XX!!??%)
OPTIONS       ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
OPTIONS       ===>
SAVED         ===>
SAVED         ===>
```

Sample partial report output:

```
CA JCLCHECK             INVOKED AT  9:56:07 AM ON FRIDAY NOVEMBER 5, yyyy
SYSTEM  INFORMATION: CPU MODEL=9021 SP4.3.0 TSO2.04.0 SMS1110 ENVIRONMENT ARRAY: 10000000000000000000
CALLING  PARAMETERS: O(OPTS)
DEFAULT  PARAMETERS: ACT,SYNTAX,RUNTIME
OPTS     PARAMETERS:            CC(5)  CT    SP(RPT 65 NOALL) V PROCX
                                J F XREF PXR(RPT) SX RP ER( END)
                                SYN RUNT USER(XX!!??%)
                                SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
                                PDS INCLUDE(USER01)
 OPTIONS IN EFFECT:             ACTION CCLIST(5) CTLSCAN ERROR(END) FULLLIST OPTIONS(OPTS)
                                PDS($ 9999999999) PROCXREF PULL PXREF(RPT) RP
                                RUNTIME SECURITY(ACTIVE) SPACE(REPORT 65 NOALL) SXREF(RPT)
SECURITY OPTIONS IN EFFECT: DATASET DASDVOL MGMTCLAS STORCLAS PROGRAM(DEFINE) NO
```

```
CA                                  REPORT 6 - ERROR MESSAGES
CA JCLCHECK             SYS: xxxx     JOB: USER02          PGMR: CA01                                THURSDAY JANUARY 9, yyyy
-------------------------------------------------------------------------------------------------------------------------------
STMT    MESSAGE
NUM     NUMBER     MESSAGE TEXT
-------------------------------------------------------------------------------------------------------------------------------

 1.     CAY6326S   'VSIGNON' UNKNOWN RETURN CODE 16 FROM CAISSF, SECURITY VALIDAT
        CAY6323I   FEEDBACK= 'CAS9TSS: UNKNOWN FUNCTION NOT SUPPORTED'
 7.     CAY6025E   REQUIRED PGM OR PROC PARAMETER MISSING
        CAY6037E   UNKNOWN KEYWORD 'PPM' SPECIFIED IN OPERAND OF 'EXEC'
 9.     CAY6186E   GDG INDEX NOT BUILT
        CAY6079E   DATA SET 'USER02.R60JCK.BKUP(0)' SPECIFIED AS OLD OR SHR, BUT
CAY6000 3 STATEMENTS FLAGGED IN JOB 'USER02' MAXIMUM SEVERITY WAS 12
```

Note: For information about security, see Special Usage Considerations in the *Programming Guide.* Also, see Using CA JCLCheck and CA JCLCheck Runtime Options in the *Command Reference Guide*.

Alternate User ID

The Alternate User ID feature allows for CA JCLCheck validation to be performed under the user ID that the job is run under. CA JCLCheck supports an alternative user ID by specifying an ID in the JCL being processed, or by specifying the CA JCLCheck USER() option. CA JCLCheck uses the specified user ID to validate the IDs access to a data set. However, the specified user ID is not used when accessing the contents of a file, for example, when PROCLIBs or control card files are processed. This feature extends this capability by running under the specified user ID when accessing these files.

Setup

Two CA JCLCheck options are available to control the use of the alternate user ID feature, they are SIGNON and NOSIGNON.

SIGNON uses the current user ID and makes that ID the current user for the processing duration of the job. You can establish the user ID in the following ways:

- You can specify the USER option to CA JCLCheck. When the USER option is present, this ID applies to all jobs processed in this CA JCLCheck run. The ID also overrides any user IDs that are specified in the JCL being validated.
- You can define user IDs in the JCL being validated. The most common is the specification of USER= on the JOB card. If you are using the ACF2 option, the ID can also come from one of the ACF2 specific comment cards, such as //*LOGONID user_id.

Each time a JOB card is processed, the user changes to the current user ID when it has changed. The job is validated using the ID that CA JCLCheck originally ran with when the USER option is not in use and the JCL contains no user ID. When all jobs have been processed, the ID is re-established to the original user.

You can use the SIGNON option only when CA JCLCheck is APF authorized.

NOSIGNON prevents the SIGNON option from being used. If NOSIGNON is specified, it overrides the SIGNON specified in JCKDFLT or at execution.

Note: Placing NOSIGNON in the JCLDFLT does not prevent use of the SIGNON.

For data centers that want to prevent the use of SIGNON, it is recommended SIGNON be added to the CAZ2JOEL. For more information, see the Installation Considerations (see page 62).

Example: Validate JCL that is Run Under a Different User ID

In the following JCL, USER1 wants to validate JCL that is run under the ID of USER2. The JCL specifies USER2 on the JOB card USER= parameter. Typically, USER1 does not have access to the library specified in the JCLLIB statement, which contains the VALPROC procedure. The library specified in the SYSIN statement is also not typically accessible by USER1.

```
//TESTJOB JOB …,
//  USER=USER2
//PROC     JCLLIB ORDER=DEV.DEPT.TEST.PROCLIB
//JS10     EXEC  VALPROC
//SYSPRINT  DD  SYSOUT=*
//SYSUT3    DD  UNIT=VIO,SPACE=(CYL,(2,1))
//SYSUT4    DD  UNIT=VIO,SPACE=(CYL,(2,1))
//SYSIN     DD  DSN=USER2.JCL.CNTL(XXX),DISP=SHR
```

The following JCL shows the results when the SIGNON option is enabled. The JOB card contains a USER=USER2 parameter. The CAY6636I message indicates that the user ID changed to USER2. No error messages occur for the PROCLIB or SYSIN control cards.

```
//TESTJOB JOB  …,
//  USER=USER2
CAY6636I SWITCHING SECURITY USERID TO 'USER2'
CAY6111I JOBNAME ON JOBCARD 'TESTJOB' DOES NOT MATCH PDS MEMBER NAME
         'SECJOB2'

CAY6000 1 STATEMENTS FLAGGED IN JOB 'TESTJOB' MAXIMUM SEVERITY WAS 0
CAY6636I SWITCHING SECURITY USERID TO 'USER1'
```

The following JCL shows the results when the SIGNON option is not enabled.

```
//TESTJOB JOB …,
//  USER=USER2
CAY6111I JOBNAME ON JOBCARD 'TESTJOB' DOES NOT MATCH PDS MEMBER NAME
         'SECJOB2'

//PROC     JCLLIB ORDER=DEV.DEPT.TEST.PROCLIB
CAY6329E ACCESS DENIED TO 'DEV.DEPT.TEST.PROCLIB' BY SECURITY, RC = 8 ACCESS
         LEVEL =  READ FOR ACID = 'USER1'

//JS10     EXEC   VALPROC
CAY6027E PROCEDURE 'VALPROC' NOT FOUND

00201005//SYSIN     DD  DSN=USER2.JCL.CNTL(XXX),DISP=SHR
CAY6329E ACCESS DENIED TO 'USER2.JCL.CNTL' BY SECURITY, RC = 8 ACCESS
         LEVEL =  READ FOR ACID = 'USER1'
CAY6329E ACCESS DENIED TO 'USER2.JCL.CNTL' BY SECURITY, RC = 8 ACCESS
         LEVEL =  READ FOR ACID = 'USER1'

CAY6000 4 STATEMENTS FLAGGED IN JOB 'TESTJOB' MAXIMUM SEVERITY WAS 8
```

Installation Considerations

The following considerations are for installation:

- CA JCLCheck must reside in an authorized load library for the SIGNON feature to be available.
- By default, the SIGNON option is included in the CAZ2JOEL (CA JCLCheck Option Exclusion List) table. If you want to use the SIGNON option, do the following steps:
  1. Remove SIGNON from the CAZ2JOEL table.
  2. Use CAZ2JCL member MZ2C049 to update the CAZ2JOEL table.
  3. Add the bolded line ($OEL SIGNON) to the IEBUPDTE input.

```
./ CHANGE NAME=CAZ2JOEL
*         $OEL  SMS                (SMS OPTION)
*         $OEL  NORUNT             (NORUNTIM OPTION)
*         $OEL  SIGNON             (SIGNON OPTION)
./ ENDUP
```

  4. Run MZ2C049.

Using the JCLNeat Component to Reformat JCL

The JCLNeat component of CA JCLCheck reformats JCL according to your specifications. This further enables you to standardize JCL processing.

Note: JCLNeat must be installed at your site. For information on installing JCLNeat, see Install JCLNeat ISPF Interface and Install JCLNeat Tables for Modification in the *Installation Guide*. Also, see Installation Options in the *Programming Guide*.

Use the instructions in the following section to perform these tasks:

- View the JCLNeat Options Panel (see page 63)
- Re-invoke the JCLNeat Options Panel (see page 64)
- View the Results (see page 64)

View the JCLNeat Options Panel

1. Select some JCL in ISPF edit, enter **%JCKNSPF** on the COMMAND line, and press enter.
2. In the JCLNeat Options panel, set your options, enter **Y** in the BYPASS THIS SCREEN field, and press enter.

```
EDIT ----- CAI.CHQA.JCLCHECK.TESTJCL(JTEST11) - 01.15 ------ COLUMNS 001  072
  COMMAND ===> %JCKNSPF                          SCROLL ===> PAGE

000001 //JTEST111 JOB '40100000',JTEST111,CLASS=K,MSGCLASS=X,
000002 //  RESTART=(JS020.PS010)
000003 //*
000004 //* THIS TESTJOB TESTS THE VALIDITY OF VARIOUS REFERBACKS IN JCL
000005 //*
000006 //TESTPROC  PROC
000007 //PS010  EXEC PGM=IEFBR14
000008 //DD011  DD  DISP=SHR,
000009 //           DSN=CAI.CHQA.TESTJCL,
000010 //           UNIT=SYSDA,
000011 //           VOL=SER=APCM04
000012 //DD012  DD  DUMMY
000013 //TESTPROC PEND
000014 //OUT1 OUTPUT CLASS=X
```

```
---------------CA JCLNeat       - OPTIONAL Parameter (DEFAULTS)----------
COMMAND ===>
                                                                    yyyy/mm/dd
COMMANDs:  CANcel; NEXT or PREVious panel after edit, save         12:27
KEYs:      Enter to edit, save.  End to edit, save, next panel.     USER002

  BCI           Block Comment Insertion                           ===> Y
  BCS           Blank Comment Separator                           ===> N
  CDEL          Comment Delete Request                            ===> 0
  CEC/CECJ      Continuation Ending Column/Job                    ===> 35  / 35
  CSC           Continuation Start Column                 ===> 16
  DCBS          Split DCB Subparameters                           ===> N
  FCC           Fill Comment Character                            ===> -
  FORMAT        Format JCL? Y-Yes, N-No, B-Basic                  ===> Y
  ICSC          Inline Comment Start Column                       ===> 37
  JOBF          Format Job Statement                              ===> 10
  MKW/MQKW      Maximum Keywords/Qualified                        ===> 60  / 10
  OPSC          Operator Start Column                             ===> 10
  PACK          Pack Operators on Cards                           ===> 12
  REXXMEM       REXX Member Name                                  ===>
  RSYM Replace Symbolics                                          ===> N
  SECC          Start/End Comment Character                       ===> *
  SEQ1/INCR     Sequencing Start/Increment Number                 ===> 10  / 10
  SIM/REPORT    Simulate Processing/Report Format                 ===> N   / 0
```

Re-invoke the JCLNeat Options Panel

To re-invoke the JCLNeat Options Panel, when the BYPASS THIS SCREEN field is set to **Y**, enter **%JCKNSPF PANEL** on the Edit panel COMMAND line, and press enter.

```
EDIT ------ CAI.CHQA.JCLCHECK.TESTJCL(JTEST11) - 01.15 ------ COLUMNS 001 072
COMMAND ===> %JCKNSPF PANEL                                  SCROLL ===> PAGE

 000001 //JTEST111 JOB '40100000',JTEST111,CLASS=K,MSGCLASS=X,
 000002 //  RESTART=(JS020.PS010)
 000003 //*
 000004 //* THIS TESTJOB TESTS THE VALIDITY OF VARIOUS REFERBACKS IN JCL
```

View the Results

To run JCLNeat against your JCL, press enter from either the JCL edit panel or the JCLNeat Options panel, depending on whether you entered Y or N in the BYPASS THIS SCREEN field in the Options panel. View the results.

```
EDIT --- CAI.CHQA.JCLCHECK.TESTJCL(JTEST11) - 01.16 ----- LAST CC WAS  00
COMMAND ===>                                    SCROLL ===> PAGE

****************************** TOP OF DATA ******************************
  000001 //JTEST111 JOB '40100000',
  000002 //            JTEST111,
  000003 //            CLASS=K,
  000004 //            MSGCLASS=X,
  000005 //            RESTART=(JS020.PS010)
  000006 //*
  000007 //**-------------------------------------------------------------*
  000008 //**
     *
  000009 //* THIS TESTJOB TESTS THE VALIDITY OF VARIOUS REFERBACKS IN JCL
  000010 //*
  000011 //**
     *
  000012 //**-------------------------------------------------------------*
  000013 //TESTPROC PROC
  000014 //PS010  EXEC PGM=IEFBR14
  000015 //DD011  DD  DISP=SHR,
  000016 //           DSN=CAI.CHQA.TESTJCL,
  000017 //           UNIT=SYSDA,
  000018 //           VOL=SER=APCM04
  000019 //DD012  DD  DUMMY
F1=HELP    F2=SPLIT   F3=END     F4=RETURN   F5=RFIND   F6=RCHANGE
F7=UP        F8=DOWN  F9=SWAP  F10=LEFT    F11=RIGHT    F12=RETRIEVE
```

```
EDIT ---- CAI.CHQA.JCLCHECK.TESTJCL(JTEST11) - 01.16 ----COLUMNS 001 072
COMMAND ===>                                              SCROLL ===> PAGE

000020 //TESTPROC PEND
000021 //OUT1           OUTPUT CLASS=X
000022 //JS010          EXEC TESTPROC
000023 //JS020          EXEC TESTPROC
000024 //*
000025 //**-------------------------------------------------------------*
000026 //**
     *
000027 //* SHOULD GET CAY6114E ON 'VOL=REF' BECAUSE STEP SKIPPED DUE TO
000028 //* RESTART
000029 //*
000030 //**
     *
000031 //**-------------------------------------------------------------*
```

Note: For information about JCLNeat reports, see the *Command Reference Guide*.

Using CA Roscoe

You can invoke CA JCLCheck from CA Roscoe in place of, or in addition to the standard CA Roscoe JCL syntax checker. You can invoke CA JCLCheck as a subtask of CA Roscoe so that using CA JCLCheck has minimum impact on the rest of the CA Roscoe system.

In addition, you can invoke CA JCLCheck using the CA Technologies supplied CA Roscoe Program Facility (RPF).

Note: For information about installing the CA Roscoe feature, see the *Installation Guide*.

Use the instructions in the following section to perform these tasks:

- Invoke CA JCLCheck from your Active Work Space (AWS) (see page 66)
- Invoke CA JCLCheck as a Roscoe Program Facility (RPF) (see page 66)
- Invoke JCLNeat under CA Roscoe (see page 69)

Invoke CA JCLCheck from your Active Work Space (AWS)

1. Enter **JCK**, and press enter to validate the JCL in your AWS.

   The following panel is the output of the scan.

2. Enter **JCKoptions|** (with options| being the options you want to specify in addition to options already set up).

```
JCK
> APPLID(APPLROS)     USER(STB,USER002)
>
 ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+
000001 //JTEST111 JOB '40100000',JTEST111,CLASS=K,MSGCLASS=X,
000002 //  RESTART=(JS020.PS010)
000003 //*
000004 //* THIS TESTJOB TESTS THE VALIDITY OF VARIOUS REFERBACKS IN JCL
```

```
> APPLID(APPLROS)     USER(STB,USER002)
>
 ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+
==MSG> CAY6000 11 STATEMENTS FLAGGED IN JOB "JTEST111" MAXIMUM SEVERITY WAS 8
==MSG>
==MSG>
==MSG>
000001 //JTEST111 JOB '40100000',JTEST111,CLASS=K,MSGCLASS=X,
000002 //  RESTART=(JS020.PS010)
000003 //*
000004 //* THIS TESTJOB TESTS THE VALIDITY OF VARIOUS REFERBACKS IN JCL
```

```
JCK CT DEST (J) EA SEC() J F X PX(R) RP ER(E I)
> APPLID(APPLROS)     USER(STB,USER002)
>
 ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+.
000001 //JTEST111 JOB '40100000',JTEST111,CLASS=K,MSGCLASS=X,
000002 //  RESTART=(JS020.PS010)
000003 //*
000004 //* THIS TESTJOB TESTS THE VALIDITY OF VARIOUS REFERBACKS IN JCL
```

Invoke CA JCLCheck as a Roscoe Program Facility (RPF)

1. Enter **JCKRPF** and press enter to invoke the Roscoe Program Facility.

   When you invoke CA JCLCheck as an RPF, CA JCLCheck displays a series of panels like the ISPF panels. Option 1 checks the contents of the AWS. Option 2 is similar to the ISPF Option 1, allowing you to scan multiple jobs and produce reports. Option 3 allows you to select criteria for a batch run of CA JCLCheck.

2. Enter **2** in the OPTIONS field on the CA JCLCheck Roscoe menu, and press enter to invoke CA JCLCheck under CA Roscoe as Foreground (ETSO).

```
JCKRPF
> APPLID(APPLROS)     USER(STB,USER002)
>
 ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+.
```

```
========================= CA JCLCHECK/Roscoe MENU ======================
OPTION ===> 2


Date:  yyyy/mm/dd     User: USER02     Time: 10:58


     0 - OPTION SPECIFICATIONS

     1 - CA JCLCHECK - CHECK AWS (JCK)

     2 - CA JCLCHECK - FOREGROUND(ETSO)

     3 - CA JCLCHECK - BATCH CHECK

     4 - CA 7 INTERFACE INVOCATION

     5 - CA Scheduler INTERFACE INVOCATION

     X - EXIT

     PF3 - EXIT

=======================================================================
```

```
======================== CA JCLCHECK/INVOCATION MENU ====================
 OPTION SELECTED: FOREGROUND(ETSO)
   DATE: yyyy/mm/dd            USER: USER02             TIME: 13:10

 DATASET NAME ====> 'CAI.CHQA.JCLCHECK.TESTJCL'
 MEMBER       ====> JTEST11    (FULL NAME, BLANK, OR *)
 LIBRARY TYPE ====> ROS        (ROS,PDS,LIB)
 CPU-ID       ====>            (MULTI-CPU ONLY)
 BATCH JCL    ====> Y          (Y/N; EDIT JCL FOR BATCH OPTION?)
 ALLOC FILES  ====> N          (Y/N; ALLOCATE FOREGROUND FILES?)

 SPECIFY ANY ADDITIONAL OPTIONS (FOR THIS RUN ONLY) BELOW
 OPTIONS      ====> CT DEST (J) EA SEC() J F X PX(R0 RP ER(E I)
 OPTIONS      ====> SP (R 85 ALL) CC(123) SE(00) SX(R) GRAPH T LIS
 OPTIONS      ====>
 OPTIONS      ====>
 SAVED                ====>
 SAVED                ====>

  Enter - PROCESS                      PF3 - RETURN
=======================================================================
```

The following is the sample output:

```
  > APPLID(APPLROS)    USER(STB,USER02)
  >
  ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+.
JCLCheck          INVOKED AT  8:41:21 AM ON THURSDAY NOVEMBER 11, yyyy
CALLING  PARAMETERS:   O(OPTS)
OPTS     PARAMETERS:   CC(5)  CT    SP(RPT 65 NOALL) V PROCX
                       J F XREF PXR(RPT) SX RP ER( END)
                       SYN RUNT
                       SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
                       PDS INCLUDE(JTEST11)
OPTIONS IN EFFECT: ACTION CCLIST(5) CTLSCAN ERROR(END) FULLLIST OPTIONS(OPTS)
                       PDS($ 9999999999) PROCXREF PULL PXREF(RPT) RP
                       SPACE(REPORT 65 NOALL) SXREF(RPT) VSAM() XREF
SECURITY OPTIONS IN EFFECT:DATASET DASDVOL MGMTCLAS STORCLAS PROGRAM(DEFINE)NO
```

```
CA                                REPORT 1 - LISTING OF JOBSTREAM JCL
 CA JCLCHECK             SYS: xxxx     JOB: JTEST111       PGMR: JTEST111
 ------------------------------------------------------------------------
 STMT
 NUM     STATEMENT TEXT
 ------------------------------------------------------------------------
  .      .+....1....+....2....+....3....+....4....+....5....+....6....+....7....+.
  1. //JTEST111 JOB '40100000',JTEST111,CLASS=K,MSGCLASS=X,
     //    RESTART=(JS020.PS010)
```

```
  > APPLID(APPLROS)    USER(STB,USER02)
  >
  ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+..
CA                                    REPORT 6 - ERROR MESSAGES
CA JCLCHECK             SYS: xxxx     JOB: USER02      PGMR: CA01
------------------------------------------------------------------------------
STMT  MESSAGE
NUM   NUMBER     MESSAGE TEXT
------------------------------------------------------------------------------

1.    CAY6326S   'VSIGNON' UNKNOWN RETURN CODE 16 FROM CAISSF, SECURITY VALIDAT
      CAY6323I   FEEDBACK= 'CAS9TSS: UNKNOWN FUNCTION NOT SUPPORTED'
21.   CAY6148W   DSNAME 'CAI.CHQA.J661.DD021' FALLS BELOW MINIMUM SPECIFIED VAL
26.   CAY6148W   DSNAME 'CAI.CHQA.J661.DD031' FALLS BELOW MINIMUM SPECIFIED VAL
36.   CAY6352E   REFERBACK STATEMENT CANNOT FOLLOW A FORWARD REFERENCE TO THIS
46.   CAY6051E   STATEMENT REFERENCED BY BACK REFERENCE '*.DD054' CANNOT BE FOU
49.   CAY6051E   STATEMENT REFERENCED BY BACK REFERENCE '*.DD041' CANNOT BE FOU
50.   CAY6051E   STATEMENT REFERENCED BY BACK REFERENCE 'JS070' CANNOT BE FOUND
      CAY6145I   CONDITION CODE PROCESSING TERMINATED DUE TO ERRORS
55.   CAY6114W   STATEMENT 9 REFERENCED BY BACK REFERENCE 'JS010.PS010' IS IN A
60.   CAY6051E   STATEMENT REFERENCED BY BACK REFERENCE 'JS020' CANNOT BE FOUND
76.   CAY6076E   UNIT 'SYSDA' INCONSISTENT WITH VOLUME 'ABCDEF'
      CAY6067W   UNABLE TO VERIFY THAT DATA SET 'CAI.CHQA.DD102' IS ON VOLUME'
      CAY6148W   DSNAME 'CAI.CHQA.DD102' FALLS BELOW MINIMUM SPECIFIED VALUE OF
      CAY6086E   BLKSIZE INCONSISTENT WITH LRECL FOR DATA SET WITH FIXED RECORD
86.   CAY6051E   STATEMENT REFERENCED BY BACK REFERENCE '*.OUT2' CANNOT BE FOUN
```

Invoke JCLNeat under CA Roscoe

1. Enter **JCKNRPF** on the command line. If it is the first time you are entering this command access the Options panel to establish your options.
2. To establish your options enter **Y** in the BYPASS THIS SCREEN field, and press **F3** (End) to exit this panel to save your changes. If entering the JCKNRPF command does not display the JCLNeat Options panel, specify **JCKNRPF PANEL**.

   View the results.

```
JCKNRPF
 > APPLID(APPLROS)    USER(STB,USER02)
 >
 ....+....1....+....2....+....3....+....4....+....5....+....6....+....
```

```
JCKNRPF PANEL
 > APPLID(APPLROS)    USER(STB,USER02)
 >
 ....+....1....+....2....+....3....+....4....+....5....+....6....+....
```

```
-------------CA JCLNeat       - OPTIONAL Parameter (DEFAULTS)-------------
COMMAND ===>
                                                                  yyyy/mm/dd
COMMANDs:   CANcel; NEXT or PREVious panel after edit, save.      12:27
KEYs:       Enter to edit, save.  End to edit, save, next panel.  USER002

  BCI           Block Comment Insertion                  ===> Y
  BCS           Blank Comment Separator                  ===> N
  CDEL          Comment Delete Request.                  ===> 0
  CEC/CECJ      Continuation Ending Column/Job           ===> 35    / 35
  CSC           Continuation Start Column        ===> 16
  DCBS          Split DCB Subparameters                  ===> N
  FCC           Fill Comment Character                   ===> -
  FORMAT        Deactivate JCL Formatting        ===> Y
  ICSC          Inline Comment Start Column              ===> 37
  JOBF          Format Job Statement                     ===> 10
  MKW/MQKW      Maximum Keywords/Qualified               ===> 60    / 10
  OPSC          Operator Start Column                    ===> 10
  PACK          Pack Operators on Cards                  ===> 12
  REXXMEM       REXX Member Name                         ===> N
  RSYM Replace Symbolics                                 ===> N
  SECC          Start/End Comment Character              ===> *
  SEQ1/INCR     Sequencing Start/Increment Number        ===> 10    / 10
  SIM/REPORT    Simulate Processing/Report Format        ===> N     / 0
```

Note: For information about the invocation of CA JCLCheck from CA Roscoe, see Using CA JCLCheck in the *Command Reference Guide*. For information about the use of JCLNeat under CA Roscoe, see Using JCLNeat in the *Command Reference Guide*.

Integrating Products

CA JCLCheck provides interfaces to many CA Technologies products and other products.

Use the instructions in the following section to perform these tasks:

- Set Up CA APCDOC Integration (see page 70)
- Set Up CA ASM2 Integration (see page 76)
- Set Up CA Dispatch Integration (see page 77)
- Set Up CA TLMS Integration (see page 77)
- Set Up CA Easyproclib Integration (see page 77)
- Set Up CA Scheduler Integration (see page 78)
- Set Up CA 1 Integration (see page 79)
- Set Up DB2 Integration (see page 79)
- Set Up IMS Integration (see page 79)
- Set Up CA WA CA 7 Edition Integration (see page 80)
- Set Up CA WA Restart Option for z/OS Schedulers Integration (see page 81)

Set Up CA APCDOC Integration

Note: This step includes the fictitious CA APCDOC system name of SYSTEMS. You would use your own system name that is already defined to CA APCDOC.

1. Access the CA APCDOC Main Menu (CASUGMSP), select Option **4** (Maintenance), and press enter.
2. Enter **6** (DOCSCAN) on the Maintenance (Selection) panel (CASUMMSP), and press enter.
3. On the Maintenance DOCSCAN (Selection) panel (CASUMNNP), select Option **2** to access the DOCSCAN Selection panel.
4. On the Maintenance DOCSCAN (Selection) panel (CASUMNSP), enter all pertinent information for the system you intend to check with CA JCLCheck.

```
CASUGMSP ----    CA APCDOC MAIN MENU    -    REL 1.3    ----------------
      OPTION ===> 4

  1  CROSS REF      - Display Various Cross-References
  2  DOCUMENTATION  - Display and Update Documentation
  3  REPORTS        - Produce and View Reports
  4  MAINTENANCE    - Perform Database Maintenance Functions
  5  SYSGEN         - Update SYSGEN Information
  6  CA JCLCHECK    - Invoke CA JCLCHECK

  X  EXIT           - Terminate CA APCDOC


  Please Enter HELP for Helpful Information
       GENLEVEL for Genlevel Information
```

```
CASUMMSP ---------    MAINTENANCE    (SELECTION)    ---------------------
  OPTION ===> 6


  1  DELETE         - Delete Cross-Reference Records
  2  UPLOAD         - Upload the Documentation Database
  3  DOWNLOAD       - Download the Documentation Database
  4  DBCOPY         - Copy Documentation from One Database to Another
  5  DOCPURGE       - Purge Cross-Reference or Documentation Records in Batch
  6  DOCSCAN        - Invoke DOCSCAN to Update the Cross-Reference Database
  7  DOCCHECK       - Invoke DOCCHECK to Check the Cross-Reference Records
```

```
  --------    MAINTENANCE: DOCSCAN    (SELECTION)    ---------------------
  OPTION ===>       2

  1  PROCLIB        - Update Procedure Libraries
  2  DOCSCAN        - DOCSCAN Submission
```

```
 ---------    MAINTENANCE: DOCSCAN    (SELECTION)    ---------------------
 COMMAND ===>


Input Job Stream
  Joblist Name              ==>      (Enter * for a list)
  Liblist Name              ==>      (Enter * for a list)
  Data Set Name             ==>    'CAI.CHQA.J670.CHQAJCL'
  Data Set Type             ==> P    (S=Seq | P=PDS | N=PANVALET | L=LIBRARIAN)
  Members to Include or Exclude ==>      (I=Include | X=Exclude)
    ==> APCPROD1
    ==> APCPROD2
    ==> APCPROD3
  Member Range
  From ==>To ==>

SYSID                    ==> Processing Options:       Reports to produce:
VSAM Catalogs Used  ==> Y (Y|N)  DELTA  ==> N (Y|N)  Tape Pull ==> N (Y|N)
VSAM Password           ==>    PRCDELTA ==> N (Y|N)  Report==> N (Y|N)
                               DIRUPD  ==> N (Y|N)  Job Action ==> N (Y|N)

 System Name       ==> SYSTEMS      APPEND Before     ==>   First ==>   (Y)
 Predecessors      ==>   (Y|N)      APPEND After      ==>   Last  ==>   (Y)
```

5. Enter **0** on the CA JCLCheck/SPF Menu, press enter.
6. Enter **1** on the CA JCLCheck Control Options Menu, press enter.
7. Enter **Y** in the DICT field, and press **F3** (End) to exit the panel.

```
JCK0200-------------- CA JCLCHECK/SPF MENU -------------------------
OPTION ===> 0


 Date:  yyyy/mm/dd          User: USERID1     Level: xx.x

  0 - Option Specifications

  1 - CA JCLCHECK - Foreground Invocation

  2 - CA JCLCHECK - Foreground Invocation / Submit

  3 - CA JCLCHECK - Batch Submit

  4 - CA 7 Interface Invocation

  5 - CA Scheduler Interface Invocation

  6 - JCLNeat - Batch Submit

  7 - RESOLVE - External Variable Resolution

  X - Exit
```

```
 ----------------- CA JCLCHECK/CONTROL OPTIONS MENU ------------------
 OPTION ===> 1


 Date: yyyy/mm/dd          User: USER02      Time: 10:58

 1 - Process Control Options

 2 - Output Control Options

 3 - Security Options

 4 - JCLNeat Options

 5 - Job Card Specifications

 X- Exit
```

```
 -----------CA JCLCHECK PROCESS CONTROL OPTIONS-------------------------
 COMMAND ===>

 Date: mm/dd/yyyy           User: USER02              Time: 10:58

       Go to ADDITIONAL PROCESSING OPTIONS PANEL   => N

CCLIST  - Number of program control cards to be listed         (0-999)  => 5
COND    - Simulate execution according to condition codes               => N
CTLSCAN - Scan utility control statements                       => Y
EASYPROC- CA Easyproclib support                                => N
MSS     - Stage VTOC of MSS volumes if necessary                => N
PXREF   - Check program and PDS member existence                => Y
SEV     - Minimum severity of errors to be listed              (0-16)   => 0
SPACE   - Check blksize efficiency for new DASD files                   => Y
  (ALL) - Check blksize efficiency for existing DASD files too          => N
  (NN)  - Minimum % of track usage (give warning if less)       (0-99)   => 65
SXREF   - Check JCL across jobs                                 => N
VSAM    - Process private catalogs (JOBCAT, STEPCAT)            => Y
PROC    - Define alternate proclibs                             => N
        - Alternate proclib(s) are selected                     => N
DICT    - Define CA APCDOC XREF database                        => Y
          Update PROFILE with these options                     => Y
```

8. On the Define JCLCheck/APCDOC Data Base panel (JCK08), enter the name of the system in the SYSTEM field.
9. Enter the CA APCDOC high-level qualifier in the CA APCDOC Database high-level qualifier field.
10. Enter the CA APCDOC database name in the Database Name field.

11. Enter the CA APCDOC database path in the Database Path field.

    Note: As an alternative, you can enter this information in the OPTIONS fields on the INVOCATION panel.

    Do not use the update parameter on the DICT option. CA JCLCheck obtains information about the predecessors from the database and uses it when scanning this job.

    Press **F3** (End) to exit the panel.

```
-----------------DEFINE JCLCHECK/APCDOC DATA BASE ----------------- JCK08
COMMAND ===>

DATE - yyyy/mm/dd          USER - USER02             TIME - 10:13

DICT     - Production XREF Data Base Option
UPDATE   - Update the XREF Data Base from this run      => N

SYSTEM   - System Name   => SYSTEMS

   For CA APCDOC Release 1.1 and above, DBHLQ is required:

   APCDOC Database high level qualifier => 'CAI.CHQA'

------------------------------------------------------------------------
   For CA APCDOC Release 1.0, database names are required:

   APCDOC Database Name    => 'CAI.CHQA.CAAPCDOC.SYSDICT'
   APCDOC Database Path    => 'CAI.CHQA.CAAPCDOC.SYSPATH'

   Important note:
   Enter either high level qualifer or database names, but not both.
```

12. Return to the Foreground INVOCATION panel (JCK0201) and enter the data set name from your CA APCDOC panel in the Data Set Name field.
13. Enter the library type in the Library Type field.
14. Enter the member, with a job name in the middle of the system (SYSTEMS) (see number 8 in this group of steps), in the Member field, and press enter.

    The result is a list of predecessor/successor relationships for the jobs defined to that system.

```
---------- CA JCLCHECK/INVOCATION -----------------------------
COMMAND ===>

Option Selected: FOREGROUND

Date: yyyy/mm/dd  User: USER02                      Time: 10:54

Data Set Name ===> 'CAI.APCMTL.CHQAJCL'
Library Type ===> PDS       (SEQ, PDS, PAN, LIB)
Member    ===> USER02       (Batch Submit -- Use * for all members)
Volume Serial ===>          (If Not CATALOGED)
CPU-ID    ===>              (Multi-CPU Only, blank: This CPU)
SYSTERM   ===> N            (Y/N/P, Error Statement and Messages only)
SYSPRINT  ===> Y            (Y/N/P, Reports 1-9 according to options )

Specify any additional options (FOR THIS RUN ONLY) below
OPTIONS   ===> CC(5)  CT    SP(RPT 65 NOALL) V PROCX
OPTIONS   ===> J F XREF PXR(RPT) SX RP ER( END)
OPTIONS   ===> DICT  SYN RUNT
OPTIONS   ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
OPTIONS   ===>
SAVED     ===>
SAVED     ===>
```

15. You can update the CA APCDOC database from CA JCLCheck by specifying **Y** in the Update the XREF Data Base from this run field on the Define JCLCheck/APCDOC database panel (JCK08). This is great for stand-alone jobs or when scanning an entire system.

16. Enter the library type in the Library Type field.

    Note: For information about the DICT, SYSID, and SYSTEM runtime options, see the *Command Reference Guide*.

```
---------------DEFINE JCLCHECK/APCDOC DATA BASE ----------------- JCK08
COMMAND ===>

DATE - yyyy/mm/ddUSER – USERID                      TIME - 10:13

DICT     - Production XREF Data Base Option
UPDATE   - Update the XREF Data Base from this run          => Y

SYSTEM   - System Name  =>

   For CA APCDOC Release 1.1 and above, DBHLQ is required:

APCDOC Database high level qualifier       => CAI.CAAPCDOC

------------------------------------------------------------------------
   For CA APCDOC Release 1.0, database names are required:

APCDOC Database Name      =>
APCDOC Database Path      =>

  Important note:
  Enter either high level qualifer or database names, but not both.
```

```
-------------------- CA JCLCHECK/INVOCATION --------------------------
COMMAND ===>

Option Selected: FOREGROUND

Date: yyyy/mm/dd  User: USER02                        Time: 10:54

Data Set Name ===> 'CAI.APCMTL.CHQAJCL'
Library Type ===> PDS      (SEQ, PDS, PAN, LIB)
Member    ===>             (Batch Submit -- Use * for all members)
Volume Serial ===>         (If Not CATALOGED)
CPU-ID    ===>             (Multi-CPU Only, blank: This CPU)
SYSTERM   ===> N           (Y/N/P, Error Statement and Messages only)
SYSPRINT  ===> Y           (Y/N/P, Reports 1-9 according to options )

Specify any additional options (FOR THIS RUN ONLY) below
OPTIONS   ===> CC(5)  CT    SP(RPT 65 NOALL) V PROCX
OPTIONS   ===> J F XREF PXR(RPT) SX RP ER( END)
OPTIONS   ===> DICT(UP)  SYN RUNT
OPTIONS   ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
OPTIONS   ===>
SAVED     ===>
SAVED     ===>
```

Set Up CA ASM2 Integration

This option requests support of the CA ASM2 DASD storage manager. CA ASM2 requires that CA JCLCheck be APF-authorized. To set up CA ASM2 integration:

1. Specify the **ASM2** option in the OPTIONS field on the INVOCATION panel or as one of your OPTS default options.

```
OPTIONS      ===> J F XREF PXR(RPT) SX RP ER( END)
OPTIONS      ===> DICT(UP)  SYN RUNT
OPTIONS      ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
OPTIONS      ===> ASM2
SAVED              ===>
SAVED              ===>
```

Note: For information about the ASM2 runtime option, see the *Command Reference Guide*.

Set Up CA Dispatch Integration

1. Specify the DISPATCH option, along with the SYSOUT classes that CA Dispatch controls.
2. Specify the **RPtrpt** option. The REPORT Listing tells you which SYSOUTs are controlled by CA Dispatch.

```
OPTIONS      ===> J F XREF PXR(RPT) SX RP ER( END)
OPTIONS      ===> DICT(UP)  SYN RUNT
OPTIONS      ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
OPTIONS      ===> DISPATCH(R,M,A,)
SAVED        ===>
SAVED        ===>
```

Set Up CA TLMS Integration

To set up CA TLMS Integration, specify the **TL**ms option.

```
OPTIONS      ===> J F XREF PXR(RPT) SX RP ER( END)
OPTIONS      ===> TL SYN RUNT
OPTIONS      ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
OPTIONS      ===> DISPATCH(R,M,A,)
SAVED        ===>
SAVED        ===>
```

Note: For information about the TLMS runtime option, see the *Command Reference Guide*.

Set Up CA Easyproclib Integration

To set up CA Easyproclib integration, scan a member with a PROCLIB DD by entering the **EAsyproc** option.

```
OPTIONS      ===> J F XREF PXR(RPT) SX RP ER( END)
OPTIONS      ===> DICT(UP)  SYN RUNT
OPTIONS      ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
OPTIONS      ===> EA
SAVED        ===>
SAVED        ===>
```

Note: For information about the EASYPROC runtime option, see the *Command Reference Guide*.

Set Up CA Scheduler Integration

1. Enter **5** on the OPTION line, and press enter to select the CA Scheduler Interface INVOCATION panel from the CA JCLCheck SPF/Menu.
2. Enter CA Scheduler data on the Job Scheduling Interface for CA Scheduler panel, press enter.

   This submits a batch job using the CAJUTIL0 procedure. Review the results.

```
JCK0200-------------- CA JCLCHECK/SPF MENU ------------------------
OPTION ===> 5


  Date:  yyyy/mm/dd     User: USERID1    Level: xx.x

       0  -  Option Specifications

       1  -  CA JCLCHECK - Foreground Invocation

       2  -  CA JCLCHECK - Foreground Invocation / Submit

       3  -  CA JCLCHECK - Batch Submit

       4  -  CA 7 Interface Invocation

       5  -  CA Scheduler Interface Invocation

       6  -  JCLNeat - Batch Submit

       7  -  RESOLVE - External Variable Resolution

       X  -  Exit
```

```
 -------------- JOB SCHEDULING INTERFACE FOR CA Scheduler-------------------
 COMMAND ===>
 DATE  - mm/dd/yy       USER - TIMTI01           TIME - 08:21

  Simulate From Date    ==> AUTOS        Report From Time ==>
  Simulate For          ==> 1            Report To Time   ==>
  Simulate From Time    ==>              SIMWORK          ==> N
  REVISE                ==> N            SIMVOL           ==>
  Stage JCL             ==> N            Everyday         ==> N

Select Report Names to Include:
  BACKLOG      DEVUTIL          LATEJOBS         SELECTED         UNITUTIL
  DATETRAN     EXPERROR         RESALL           SIMEXEC

 Enter Any CA Scheduler Simulate Information:
 ==>
 ==>
 ==>
 ==>
 ==>
 ==>
       Press Enter to Submit or End to Return to Main Menu
```

Set Up CA 1 Integration

To set up CA 1 integration, specify the **TMS** option.

```
OPTIONS      ===> J F XREF PXR(RPT) SX RP ER( END)
OPTIONS      ===> TM SYN RUNT
OPTIONS      ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
OPTIONS      ===> DISPATCH(R,M,A,)
SAVED        ===>
SAVED        ===>
```

Note: For information about the TMS runtime option, see the *Command Reference Guide*.

Set Up DB2 Integration

To set up DB2 integration, specify the **DB2** option.

```
OPTIONS      ===> J F XREF PXR(RPT) SX RP ER( END)
OPTIONS      ===> DB2 SYN RUNT
OPTIONS      ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
OPTIONS      ===> DISPATCH(R,M,A,)
SAVED        ===>
SAVED        ===>
```

Note: For information about the DB2 runtime option, see the *Command Reference Guide*.

Set Up IMS Integration

To set up IMS integration, specify the **IMS** option.

```
OPTIONS      ===> J F XREF PXR(RPT) SX RP ER( END)
OPTIONS      ===> IMS SYN RUNT
OPTIONS      ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
OPTIONS      ===> DISPATCH(R,M,A,)
SAVED        ===>
SAVED        ===>
```

Note: For information about the IMS runtime option, see the *Command Reference Guide*.

Set Up CA WA CA 7 Edition Integration

CA WA CA 7 Edition is supported with two options. The first option is option 4 - CA 7 Interface Invocation (described here). The second option is option 7 - RESOLVE - External Variable Resolution, which is described in About External Variables (see page 81).

The following procedure describes how to set up CA WA CA 7 Edition with option 4:

1. Enter **4** on the OPTION line, and press enter to select the CA 7 Interface INVOCATION panel.
2. Do one of the following tasks:
   - On the JOB SCHEDULING INTERFACE FOR CA 7 panel, enter all necessary CA 7 data.
   - Specify the CA7 option with an SCHD ID for a job containing #XI, #XO, #JI, and #JO statements in the CA JCLCheck INVOCATION panel, press enter. View the results.

   This example requests that CA JCLCheck process the input and include or exclude JCL based on a schedule ID of 1, as if the job were to be run on Friday, August 2, 2000, at 7:00 p.m.

   This submits a batch job to invoke the CA 7 interface SSJCK0. Review the results.

   Note: For information about the CA7 runtime option, see the *Command Reference Guide*.

```
JCK0200-------------- CA JCLCHECK/SPF MENU ------------------------
OPTION ===> 4


  Date:  yyyy/mm/dd     User: USERID1    Level: xx.x

       0  -  Option Specifications

       1  -  CA JCLCHECK - Foreground Invocation

       2  -  CA JCLCHECK - Foreground Invocation / Submit

       3  -  CA JCLCHECK - Batch Submit

       4  -  CA 7 Interface Invocation

       5  -  CA Scheduler Interface Invocation

       6  -  JCLNeat - Batch Submit

       7  -  RESOLVE - External Variable Resolution

       X  -  Exit
```

```
------------------JOB SCHEDULING INTERFACE FOR CA 7--------------------
COMMAND ===>

Date - mm/dd/yyyy            User - USER02           Time - 16:30
From Date    ==> 120195      From Time               ==> 12:00
To   Date    ==> 120295      To   Time               ==> 12:00
CA 7 OPID    ==> USER02      Password                ==>


 Enter Jobs and/or Systems to be selected:

Jobname ==> PAYROLL   System ==> PRD1
Jobname ==> GENLEDGE  System ==> PRD2
Jobname ==>           System ==>
Jobname ==>           System ==>
Jobname ==>           System ==>
Jobname ==>           System ==>
Jobname ==>           System ==>
Jobname ==>           System ==>
Jobname ==>           System ==>

     Press Enter to Submit or End to return to main menu
```

```
OPTIONS      ===> J F XREF PXR(RPT) SX RP ER( END)
OPTIONS      ===> DICT(UP)  SYN RUNT
OPTIONS      ===> SEC(DASD,STOR,MGMT,PROGRAM(DEFINE))
OPTIONS      ===> CA7(1 95214 1900)
SAVED        ===>
SAVED        ===>
```

Set Up CA WA Restart Option for z/OS Schedulers Integration

If your JCL contains a step that calls CA WA Restart Option for z/OS Schedulers, specify the CA11 option. If your JCL does not contain a CA WA Restart Option for z/OS Schedulers step, specify SUPCAT. Messages such as, DATA SET ALREADY CATALOGED, is suppressed.

Note: For information about product integration, see Using CA JCLCheck and CA JCLCheck Runtime Options in the *Command Reference Guide.* See Installation Options and Job Control Standards in the *Programming Guide.*

About External Variables

You can use CA JCLCheck to validate JCL that contains external variables. This validation is required when a scheduling product uses variables embedded in its JCL. When validating JCL that contains external variables, the following steps are required before JCL validation occurs:

1. Specify the product that CA JCLCheck calls to resolve the external variables.

2. Specify the product-specific parameters for external variable resolution in a data set. CA JCLCheck then calls the product and receives resolved JCL.

This section provides a process overview and explains the steps that are required to resolve external variables using the following methods:

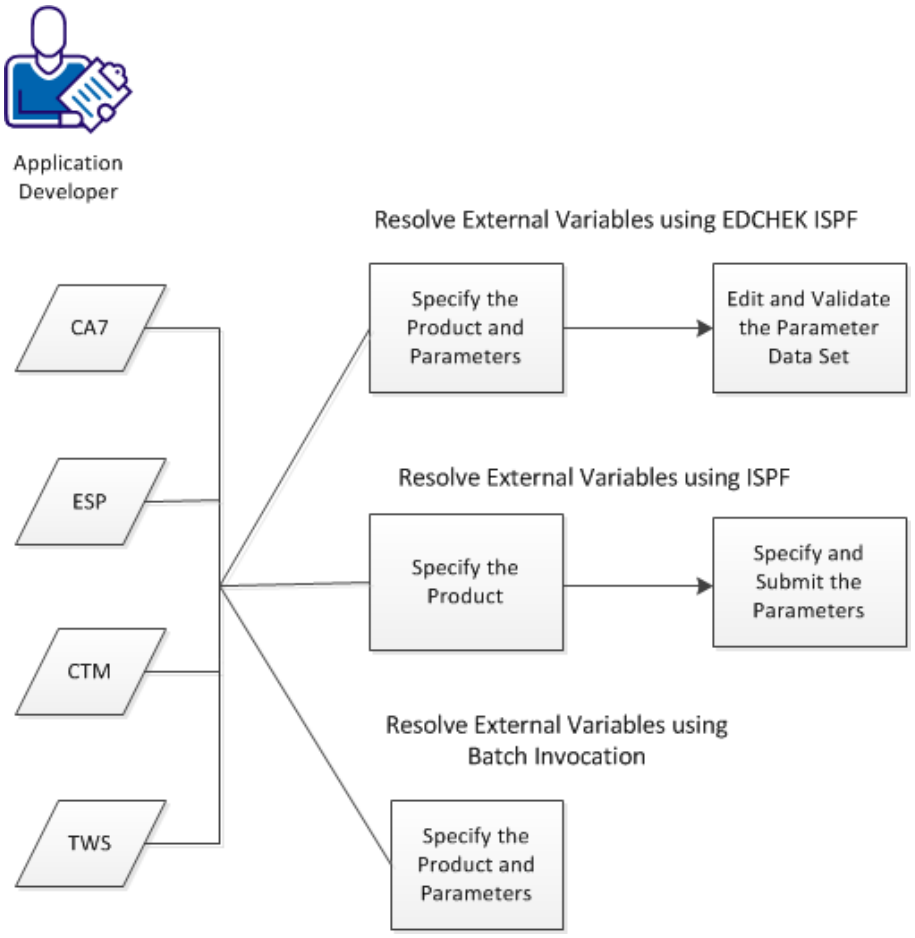
- The CA JCLCheck EDCHEK ISPF edit feature
- The CA JCLCheck ISPF panels
- Run CA JCLCheck in batch

CA JCLCheck can resolve external variables in the following scheduling products: CA WA CA 7 Edition (CA7), CA WA ESP Edition (ESP), Control-M (CTM), and Tivoli Workload Scheduler (TWS). This section also contains a description of the product-specific parameter files.

Note: For information about the installation considerations, see Resolve External Variable Installation Considerations in the *Programming Guide*.

How to Configure CA JCLCheck to Resolve External Variables

As an application developer, you are responsible for the configuration of CA JCLCheck. To configure CA JCLCheck to resolve external variables from different scheduling products, you can use three different methods. The following diagram shows the three different methods:



Follow the procedures for the method that applies to your environment.

To Resolve External Variables using EDCHEK ISPF, follow these steps:

1. Specify the Product and Parameters using EDCHEK ISPF Edit Macro (see page 84)
2. Edit and Validate the Parameter Data Set using EDCHEK ISPF Edit Macro (see page 85)

To Resolve External Variables using ISPF, follow these steps:

1. Specify the Product using ISPF (see page 86)

2. Specify and Submit the Parameters using ISPF (see page 86)

To Resolve External Variables using Batch Invocation, follow this step:

1. Specify the Product and Parameters in Batch (see page 91)

Specify the Product and Parameters using EDCHEK ISPF Edit Macro

This procedure describes how to specify the product and the product-specific parameters that CA JCLCheck calls to resolve, using the EDCHEK ISPF edit macro.

Follow these steps:

1. Enter the command %EJCK PANEL in ISPF.
2. Locate the RESOLVE field in panel JCK21.
3. Specify the scheduling product to the right of the arrow. Use one of the following scheduling product acronyms: CA7, ESP, CTM, or TWS.
4. Specify the data set name that contains the product-specific parameters in the DSN field, and optionally a member name in the MEM field.
5. Set the single-character field, found immediately before the DSN field, to 'E' to edit the parameter data set. You can also specify a 'B' to browse the data set.

The scheduling product and the product-specific parameters are now defined for CA JCLCheck.

Edit and Validate the Parameter Data Set using EDCHEK ISPF Edit Macro

This procedure describes how to edit and validate the parameter data set. If you specified an 'E' in the previous step, the contents of the parameter data set are available for editing.

Follow these steps:

1. Specify the parameters for the scheduling product.
2. Start the parameters in column 1 and stop on or before column 71.
3. Specify the parameters in any order (there is no correct parameter order).

   Note: Depending on the product, some parameters can be required. Refer to the following product parameters for details.
4. Specify comment lines by placing an asterisk (*) in column 1. Do not add comments to the right side of the parameter value.
5. Press PF3 (END) out of the parameter file.

CA JCLCheck calls the scheduling product that was specified to resolve its variables. CA JCLCheck then receives a resolved JCL deck and validates the resolved JCL.

After the JCL is validated, you are returned to the ISPF edit session showing ==MSG> lines pointing to JCL lines in error. The error lines are resolved JCL lines and they can look different from the unresolved JCL lines in the ISPF edit session. Also, since the resolved JCL could have JCL lines that were added or deleted, error messages can appear for JCL lines not present in the unresolved JCL deck. Note the differences between the unresolved and resolved JCL.

For a list of the parameters for each supported scheduling product, see the following topics:

- CA WA CA 7 Edition Parameters (see page 95)
- CA WA ESP Edition Parameters (see page 100)
- Control-M Parameters (see page 104)
- Tivoli Workload Scheduler Parameters (see page 107)

Specify the Product using ISPF

This procedure describes how to specify the product and the product-specific parameters that CA JCLCheck calls to resolve, using ISPF.

Follow these steps:

1. Select option number 0 (zero) on the CA JCLCheck ISPF main menu panel (JCK0200).
2. Select option number 1 on the Control Options Menu panel (JCK0210).
3. Specify a 'Y' in the 'Go to ADDITIONAL PROCESSING OPTIONS PANEL' field on the Process Control Options panel (JCK0202).
4. Specify the scheduling product in the RESOLVE field on the Additional Processing Options panel (JCK0203). Use one of the following scheduling product acronyms: CA7, ESP, CTM, or TWS.
5. Return to the CA JCLCheck ISPF main menu panel (JCK0200). The scheduling product is now defined for CA JCLCheck.
6. Select option number 7 - RESOLVE - External Variable Resolution to begin the submission process.

Specify and Submit the Parameters using ISPF

This procedure describes how to specify the product-specific parameters that CA JCLCheck calls to resolve, using ISPF. If you returned to the CA JCLCheck ISPF main menu panel and selected option 7 in the previous step, the External Variable Specification panel is available. A different panel is in use for each scheduling product; this topic contains an example of each panel. For a list of the parameters for each supported scheduling product, see the following topics or the Tutorial Help panels:

- CA WA CA 7 Edition Parameters (see page 95)
- CA WA ESP Edition Parameters (see page 100)
- Control-M Parameters (see page 104)
- Tivoli Workload Scheduler Parameters (see page 107)

Follow these steps:

1. Specify all relevant parameters for the scheduling product in the products panel.
2. Specify the name of the resolve parameter data set in the 'Resolve Parm File' field.

   The resolve parameter data set must be a sequential file. If no data set name is specified, userid.JCKRESLV.PARMS is used. If the Resolve Parm File does not exist, it is allocated automatically.
3. Specify the Input File to identify the JCL that is going to be resolved.

4. Specify the file allocation size for the RESJCL file that is used to pass the resolved JCL to CA JCLCheck for validation. This file is temporary.
5. Enter a 'Y' in the 'Update PROFILE with these options' field and hit enter.
6. Depending on the scheduling product, the Resolve Parm File may display in edit mode so you can enter options. After editing, Press PF3 (END) to save.

```
EDIT       USERID1.SPF.RESPARMS                        Columns 00001 00072
Command ===>                                             Scroll ===> PAGE
****** **************************** Top of Data ************************
000100 LIBRARY  USERID1.JCL.CNTL
000200 MEMBER   CTMJCL
000300 SUBMIT
000400 MEMBER   CTMJCL2
000500 SUBMIT
****** ************************ Bottom of Data **************************
```

7. The External Variable Resolution Submission panel (JCK0217) displays for all scheduling products.
8. Specify the following fields: Foreground or Batch submit, SYSTERM, SYSPRINT, and any additional CA JCLCheck options. The following are sample options:

```
------------ CA JCLCheck External Variable Resolution Submission -------------
COMMAND ===>

 DATE: yyyy/mm/dd                  USER: CAUSR01                   TIME: hh:mm

   Foreground or Batch submit  => B   (F/B)
   CPU-ID        ===>            (Multi-CPU Only, blank: This CPU)
   SYSTERM       ===> Y          (Y/N/P, Error Statement and Messages only)
   SYSPRINT      ===> Y          (Y/N/P, Reports 1-11 according to options)

Specify any additional options (FOR THIS RUN ONLY) below
OPTIONS => CC(100)  NOCT    NOSPA V  NOPROCX
OPTIONS => FULL NOPXR NOSX RP ER( END)
OPTIONS => SYN RUNT
OPTIONS => J XREF   NOAUD NOREM NOREXXMS
OPTIONS => NOMCOUSR NOMCOSYS
OPTIONS => NOCHA
OPTIONS =>
OPTIONS =>
SAVED   =>
SAVED   =>
```

Once you set any field, it is set to the same value when you return to the panel.

All other parameters on this menu are the same as the other CA JCLCheck invocation panels. When you press Enter, CA JCLCheck is invoked.

If you selected the foreground option and specified 'Y' in the SYSPRINT field, you see the SYSPRINT file from the preprocessor of the scheduling product. This SYSPRINT file is in addition to the standard CA JCLCheck SYSPRINT file.

Example: Specify Parameters in the External Variable Specification Panel

CA WA CA 7 Edition

This example shows the CA WA CA 7 Edition External Variable Specification panel that is used when specifying parameters using ISPF:

```
JCK0218 ------ CA JCLCheck External Variable Specification - CA 7 -------------
COMMAND ===>
DATE: yyyy/mm/dd                  USER: USERID1                   TIME: hh:mm

Source             ===> E               (E-External, I-Internal; Required)
Source=E fields:
  Input File       ===> USERID1.JCK.CNTL(PYRLTEST)
  Work JCLLIB      ===> &WORKJCL        (Required)
    Member Name    ===> PAYROLL9        (Required)
Source=I fields:
  Job Name         ===>                 (Required)
  Prod JCLLIB      ===>                 (Optional)
Date, Time, SCHID  ===> 11353 (yyddd)   ===> 2000 (hhmm)    ===> 1   (nnn)
CA 7 CCI Node      ===> A31SENF         CA 7 Instance  ===> C377
Resolve Parm File  ===> USERID1.JCKRESLV.PARMS
RESJCL File Size   ===> 1     ===> 1     (Primary, Secondary Cylinders)
Debugging Control  ===>                 (R-Record, S-RecordSeq, RS, or blank)

Update PROFILE with these options  ===> Y

                    Press Enter to continue
```

CA WA ESP Edition

This example shows the CA WA ESP Edition External Variable Specification panel that is used when specifying parameters using ISPF:

```
JCK0209 ------ CA JCLCheck External Variable Specification - ESP --------------
COMMAND ===>
DATE: yyyy/mm/dd                  USER: USERID1                   TIME: hh:mm
SCAN or SIMULATE   ===> SIMULATE   (Required)
ESP Subsystem Name ===> ES51       SYMBOL ===>
Input File    ===> CYB1.ESPJCK.CNTL(B)
EVENT         ===> USERID1.APPL1
SCHED         ===>
SYMLIB        ===> USERID1.JCK.CNTL(SYMLIB)
For SCAN only:
  CALENDARS   ===>
For SIMULATE only: (EVENT and PROCJOB are Required)
  PROCJOB     ===> IDCAMS.TEST
  VARS        ===>
  USER1       ===> IEFBR14
  USER2       ===>
  USER3       ===>
  USER4       ===>
  JCLSCAN     ===> CYBJCSCN       Continue on another panel  ===> N  (Y/N)
Resolve Parm File  ===> USERID1.JCKRESLV.PARMS
RESJCL File Size   ===>       ===>       (Primary, Secondary Cylinders)
Debugging Control  ===> RS               (R-Record, S-RecordSeq, RS, or ' ')

Update PROFILE with these options  ===> Y   Press Enter to continue
```

Tivoli Workload Scheduler

This example shows the Tivoli Workload Scheduler External Variable Specification panel that is used when specifying parameters using ISPF:

```
JCK0207 ------ CA JCLCheck External Variable Specification - TWS --------------
COMMAND ===>
DATE: yyyy/mm/dd                  USER: USERID1                    TIME: hh:mm
   TWS Subsystem Name  ===> OPOP
   Source              ===> EI               (E-External, I-Internal, or EI)
   Input File          ===> USERID1.JCK.CNTL
     Volume Serial     ===>                  (If not cataloged)
     Input File Type   ===> SEQ              (SEQ, PDS, PAN, LIB)
     Required if Input File Type = SEQ:
       WSname  ===> WSPR    OPNO  ===> 10    Jobname  ===> USERID1J
   Application ID      ===> TWSAPPLICATION   (Required)
   Valid Date          ===> 100721           (YYMMDD)
   IA Date/Time        ===> 1004301800       (YYMMDDHHMM)
   Work Application    ===>                   OR POSITION =>   TEXT =>
   Work Group ID       ===> GROUPE
   Work Owner ID       ===> TST-OW1
   Simulate Date/Time  ===>                  (YYYYMMDDHHMM)
   Variable Table      ===>                  (Override variable table name)
   Resolve Parm File   ===> USERID1.JCKRESLV.PARMS
   RESJCL File Size    ===> 5     ===> 1     (Primary, Secondary Cylinders)
   Replace &OADID      ===> &TJCG0
   Debugging Control   ===>                  (S-Snap, R-Record, SR, or blank)
   Update PROFILE with these options    ===> Y
   Appl Run Dates      ===>     (Y or N)     Press Enter to continue
```

Control-M

This example shows the Control-M External Variable Specification panel that is used when specifying parameters using ISPF:

```
JCK0208 --- CA JCLCHECK EXTERNAL VARIABLE SPECIFICATION - CONTROL-M ----- COMMAND ===>
 DATE: yyyy/mm/dd                 USER: USERID1                  TIME: hh:mm


    Resolve Parm File    ===> USERID1.SPF.RESPARMS
    RESJCL File Size     ===> 10    ===> 10    (Primary, Secondary Cylinders)


    Update PROFILE with these options    ===> Y
```

Pressing enter on this panel allows you to edit the Resolve Parm File. After editing, press PF3 (END) to save.

```
EDIT       USERID1.SPF.RESPARMS                          Columns 00001 00072
Command ===>                                               Scroll ===> PAGE
****** ***************************** Top of Data ************************
000100 LIBRARY  USERID1.JCL.CNTL
000200 MEMBER   CTMJCL
000300 SUBMIT
000400 MEMBER   CTMJCL2
000500 SUBMIT
****** ************************ Bottom of Data **************************
```

Specify the Product and Parameters in Batch

The following procedures describe how to specify the product and the product-specific parameters in a batch job.

The following procedure is for CA WA CA 7 Edition.

When using a batch job to invoke CA JCLCheck, it is recommended that you use the procedure CAZ1JCC7.

Follow these steps:

1. Customize the CAZ1JCC7 procedure for your environment before running for the first time.
2. Create a Resolve parameter file and edit it to specify the CA WA CA 7 Edition parameters that are required to process the JCL.

3. Create JCL to run the CAZ1JCC7 procedure. The following JCL shows how to invoke the procedure:

```
//CHECK    EXEC CAZ1JCC7,REGION=0K,
//   RESPARM='RESOLVE(CA7,dsname,member)',
//   OPTION='{JCLCheck options}'
//SYSIN    DD DISP=SHR,DSN=sequential.jcl.file (optional)
```

4. Specify the CA7 scheduling product acronym in the PARM keyword.
5. Specify a data set (and member) name that contains the CA WA CA 7 Edition parameters in the RESPARM keyword. If the data set is a sequential file, omit the member name. If the data set is a PDS, the member name must be specified. The RESPARM= parameter must contain a RESOLVE parameter. You can also specify DDNAME:ddname instead of data set and member to reference a ddname that is specified in your JCL.
6. Specify OPTION= parameter to set standard CA JCLCheck processing options. When using the CAZ1JCC7 procedure, do not specify options PDS, PANVALET, LIBRARIAN, INCLUDE, or EXCLUDE as these cause errors. CA JCLCheck processes the resolved JCL as a sequential file.
7. If you are specifying SOURCE=E in the RESPARMS options, make sure that SYSIN DD points to a sequential file containing the JCL to process. This file is not used if SOURCE=I.

The following procedure is for CA WA ESP Edition.

When using a batch job to invoke CA JCLCheck, it is recommended that you use the procedure CAZ1JCES.

Follow these steps:

1. Customize the CAZ1JCES procedure for your environment before running for the first time.
2. Create a Resolve parameter file and edit it to specify the CA WA ESP Edition parameters that are required to process the JCL.
3. Create JCL to run the CAZ1JCES procedure. The following JCL shows how to invoke the procedure:

```
//CHECK    EXEC CAZ1JCES,REGION=0K,
//   RESPARM='RESOLVE(ESP,dsname,{member}){,JOBNAME(jobname)}'
//   OPTION='{JCLCheck options}'
//SYSIN    DD DISP=SHR,DSN=sequential.jcl.file (optional)
```

4. Specify the ESP scheduling product acronym in the PARM keyword.

5. Specify a data set (and member) name that contains the CA WA ESP Edition parameters in the RESPARM keyword. If the data set is a sequential file, omit the member name. If the data set is a PDS, the member name must be specified. The RESPARM= parameter must contain a RESOLVE parameter. You can also specify DDNAME:ddname instead of data set and member to reference a ddname that is specified in your JCL. Use the JOBNAME parameter only when you are not specifying the PROCJOB parameter.

   Note: For information about the parameters, see CA WA ESP Edition Parameters (see page 100).

6. Specify OPTION= parameter to set standard CA JCLCheck processing options. When using the CAZ1JCES procedure, do not specify options PDS, PANVALET, LIBRARIAN, INCLUDE, or EXCLUDE as these cause errors. CA JCLCheck processes the resolved JCL as a sequential file.

7. If you are specifying SOURCE=E in the RESPARMS options, make sure that SYSIN DD points to a sequential file (or PDS with the member specified) containing the JCL to process. This file is not used if SOURCE=I.

The following procedure is for the Tivoli Workload Scheduler.

When using a batch job to invoke CA JCLCheck, it is recommended that you use the procedure CAZ1JCTW.

Follow these steps:

1. Customize the CAZ1JCTW procedure for your environment before running for the first time.

2. Create a Resolve parameter file and edit it to specify the TWS parameters that are required to process the TWS JCL.

3. Create JCL to run the CAZ1JCTW procedure. The following JCL shows how to invoke the procedure:

```
//CHECK    EXEC CAZ1JCTW,REGION=0K,
//   RESPARM='RESOLVE(TWS,dsname,{member}){,JOBNAME(jobname)}',
//   OPTION='{JCLCheck options}'
//SYSIN    DD DISP=SHR,DSN=tws.input.library
```

4. Specify the TWS scheduling product acronym in the RESOLVE keyword of the RESPARM EXEC Parm.

5. Specify a data set (and member) name of the Resolve parameter file in the RESOLVE keyword of the RESPARM EXEC Parm. If the data set is a sequential file, omit the member name. If the data set is a PDS, the member name must be specified. The RESPARM= parameter must contain a RESOLVE parameter and can optionally contain a JOBNAME parameter. You can also specify DDNAME:ddname instead of data set and member to reference a ddname that is specified in your JCL.

6. Specify OPTION= parameter to set standard CA JCLCheck processing options. When using the CAZ1JCES procedure, do not specify options PDS, PANVALET, LIBRARIAN, INCLUDE, or EXCLUDE as these cause errors. CA JCLCheck processes the resolved JCL as a sequential file.
7. If you are specifying SOURCE=E or IE in the parameter file, SYSIN DD points to a PDS, CA Librarian, CA Panvalet, or sequential file containing the JCL to process. The SRCETYPE keyword in the parameter file defines the type of file that is specified in the SYSIN DD. This file is not used if SOURCE=I.
8. Add the jobname to the PARM keyword when the Input File Type is SEQ (SRCETYPE=SEQ). For sequential input, the jobname field is used to specify the member name being processed.

The following procedure is for Control-M.

When using a batch job to invoke CA JCLCheck, it is recommended that you use the procedure CAZ1JCCM provided with this feature.

1. Customize the CAZ1JCCM procedure for your environment before running for the first time.
2. Create a Resolve parameter file and edit it to specify the Control-M parameters that are required to process the JCL.
3. Create JCL to run the CAZ1JCCM procedure. The following JCL shows how to invoke the procedure:

```
//CHECK    EXEC CAZ1JCCM,REGION=0K,
//   RESPARM='{RESOLVE(CTM,parameters)}{,JOBNAME(name)}',
//   OPTION='{JCLCheck options}'
//SYSIN DD DISP=SHR,DSN=sequential.jcl.file (optional)
//RESPARMS DD DSN=ctm.resparm,DISP=SHR   (optional)
```

4. The EXEC statement points to a procedure that resolves CTM variables and then calls CA JCLCheck to validate the JCL.
5. Specify a data set (and member) name of the Resolve parameter file in the RESOLVE keyword of the RESPARM EXEC Parm. If the data set is a sequential file, omit the member name. If the data set is a PDS, the member name must be specified. The RESPARM= parameter must contain a RESOLVE parameter and can optionally contain a JOBNAME parameter. You can also specify DDNAME:ddname instead of data set and member to reference a ddname that is specified in your JCL.
6. The OPTION= parameter can be used to specify standard CA JCLCheck processing options. When using the CAZ1JCCM procedure, do not specify options PDS, PANVALET, LIBRARIAN, INCLUDE, or EXCLUDE as these cause errors. CA JCLCheck processes the resolved JCL as a sequential file.

7. If no RESPARM= parameter is specified, the default value that is provided in the CAZ1JCCM procedure is:

   ```
   RESPARM='RESOLVE(CTM,DDNAME:RESPARMS)'
   ```

   The preceding statement uses the DDNAME format of the RESOLVE parameter to indicate that the Resolve Parameters are specified using the DDNAME of RESPARMS.

For Control-M, three formats are available for the RESOLVE parameter.

1. The following format allows for the use of parameters SOURCE=E, CTMPRINT and RECORD without having a parameter file defined. The SOURCE=E parameter requires the unresolved JCL to be in a sequential file pointed to by the SYSIN DD statement. The RECORD option is only used if a RECORD DD is present. The JOBNAME parameter allows you to specify the library member name that is used when calling CTMAESIM. Some variables in the JCL can use the member name in building values. If JOBNAME is not specified, the default is JCKDFLT.

   ```
   RESPARM='RESOLVE(CTM,NOPARM),{JOBNAME(member)}'
   ```

2. The following format allows for the DD name for the Resolve Parameters to be specified. The DD name that is provided must be defined in your JCL.

   ```
   RESPARM='RESOLVE(CTM,DDNAME:ddname),{JOBNAME(member)}'
   ```

3. The following format allows for the file name for the Resolve Parameters to be specified. If the file is a PDS(/e), a member name can also be specified. The file is dynamically allocated.

   ```
   RESPARM='RESOLVE(CTM,file.name(,member}),{JOBNAME(member)}'
   ```

The following rules apply when a Resolve parameter is specified:

- When the parameter file contains a SOURCE=E statement, you can use the JOBNAME parameter to specify the library member name. The library member name is used when calling CTMAESIM. The JOBNAME parameter is used only when SOURCE=E is specified. SOURCE=E requires the unresolved JCL to be in a sequential file pointed to by the SYSIN DD statement.
- When SOURCE=E is not specified, then the file the ddname references must contain CTMAESIM options. For example, Library and Member. These options identify the location of the JCL.

CA WA CA 7 Edition Parameters

This topic contains the CA WA CA 7 Edition parameters and the following example:

- Sample Parameter Data Set (see page 100)

This procedure describes how to specify the parameters.

Follow these steps:

1. Specify the CA WA CA 7 Edition parameters in a sequential file (including partitioned data set member).
2. Start the parameters in column 1 and stop on or before column 71.
3. Specify the parameters in any order (there is no correct parameter order).
4. Specify comment lines (if necessary) by placing an asterisk (*) in column 1. Do not add comments to the right side of the parameter value.

The following parameters are for CA WA CA 7 Edition:

SOURCE=*s*

Identifies the source of the job and is mandatory.

SOURCE=E means that the job is not necessarily defined to CA WA CA 7 Edition. The SOURCE=E-related fields specify which file to run variable resolution against and to which CA WA CA 7 Edition library to copy the input JCL to before resolution. When validating JCL under EDCHEK, SOURCE=E must be specified.

SOURCE=I means that the job is defined to CA WA CA 7 Edition. The actual JCL resides in a CA WA CA 7 Edition library. The user can override the library defined in the job definition by specifying the P_JCLLIB field.

Example: SOURCE=E

W_JCLLIB=*index*

Identifies the CA WA CA 7 Edition library the input file is copied to before resolution. You can specify this parameter as a symbolic (&aaaaa) or numeric (nnn) index. This parameter is mandatory when SOURCE=E.

Example: W_JCLLIB=&WORKJCL

MEMBER=***member***

Identifies the member name (in W_JCLLIB) to copy the input file to before resolution. This parameter is mandatory when SOURCE=E.

Example: MEMBER=PAYROLL9

JOB=*jobname*

Identifies the job name that is defined to CA WA CA 7 Edition, which CA JCLCheck validates. This parameter is mandatory when SOURCE=I.

Example: JOB=PAYROLL

P_JCLLIB=*index*

When SOURCE=I and the job is defined to CA WA CA 7 Edition, you can specify this parameter to override the library defined in the job definition. This parameter can be specified as a symbolic (&aaaaa) or numeric (nnn) index. This parameter is optional.

Example: P_JCLLIB=100

DATE=*yyddd*

Identifies the Julian date used in evaluating scheduled overrides (for example, #JI and #JO statements). This parameter is optional.

Example: DATE=11353

TIME=*hhmm*

Identifies the time used in evaluating scheduled overrides (for example, #JI and #JO statements). This parameter is optional.

Example: TIME=2000

SCHID=*schid*

Identifies the Schedule ID used in evaluating scheduled overrides (for example, #JI and #JO statements). This parameter is optional.

Example: SCHID=1

NODE=*node*

Identifies the CA WA CA 7 Edition CCI terminal interface node. If omitted, the default is the local node.

Example: NODE=A31SENF

INSTANCE=*instance*

Identifies the CA WA CA 7 Edition instance name. If omitted, the default is CA71.

Example: INSTANCE=C377

RECORD

Informs CA JCLCheck to record the resolution session. The record data is written to the partitioned data set allocated in the RECORD DD statement. Allocate this file as a PDS whose DCB is LRECL=500, RECFM=FB, and DSORG=PO. Under EDCHEK and ISPF, the name of this file must be specified in the JCKUSRC7 CLIST. In a batch job, the JCL deck must contain a RECORD DD statement.

Use this parameter only under the direction of CA Support.

Example: RECORD

RECORDSEQ

Informs CA JCLCheck to record the resolution session. The record data is written to a sequential data set allocated in the RECORDSQ DD statement. Allocate this file as a sequential file whose DCB is LRECL=500, RECFM=FB. Under EDCHEK and ISPF, the name of this file must be specified in the JCKUSRC7 CLIST. In a batch job, the JCL deck must contain a RECORDSQ DD statement.

Use this parameter only under the direction of CA Support.

Example: RECORDSEQ

CA WA CA 7 Edition Driver Procedures

Read this section if the JCL being resolved contains a reference to a Driver Procedure (DPROC).

If your site uses only the global DPROC library (CARPROC DD statement in the CA7ONL job), JCL with references to a DPROC contained in the global DPROC library are resolved.

If your site uses both global DPROCs and local DPROCs, CA WA CA 7 Edition looks for the DPROC referenced in the JCL in the local DPROC first. A local DPROC is associated with a specific JCLLIB. If the DPROC is not found, CA WA CA 7 Edition then looks in the global DPROC. The DPROC remains unresolved when it is not found in either location.

All JCLLIB information applies equally whether you specify SOURCE=I or you specify SOURCE=E and W_JCLLIB=. When you specify SOURCE=I, the JCLLIB containing the JCL of the job is probably already associated with the local DPROC library (unless your site uses only global DPROCs). However, when you specify SOURCE=E, do the following steps:

1. Associate the W_JCLLIB used with a local DPROC library.
2. Copy the actual DPROCs to the local DPROC library (unless the DPROC referenced in your JCL resides in the global DPROC library).

   This association and copy ensure that the DPROCS referenced in the JCL are resolved.

Example: Code JCL to Reference a Local and Global DPROC

This example shows how to refer to both a local (NMDPROCL) and global (NMDPROCG) DPROC:

```
//JOBCA7  JOB
/*JOBPARM S=*
//*** comments and other JCL statements
//        NMDPROCG,DATA1=data,DATA2=data
//JS020   EXEC PGM=IEFBR14
//        NMDPROCL,PARM1=data,PARM2=data
```

Let us assume that this JCL is coming from a CA WA CA 7 Edition JCL library, &JCLLIB1. &JCLLIB1 has defined a local DPROC library (HLQ.DPROC.LOCAL) and member NMDPROCL resides in that library. In the global DPROC library, which is coded on the CARPROC DD statement in the CA7ONL job, we have many members. One of the members is NMDPROCG. When CA WA CA 7 Edition encounters the following statement, it looks in the local DPROC library (HLQ.DPROC.LOCAL) and does not find NMDPROCG.

```
//     NMDPROCG,DATA1=data,DATA2=data
```

CA WA CA 7 Edition then looks in the CARPROC DD data set and finds the member NMDPROCG, so CA WA CA 7 Edition resolves the statements.

As CA WA CA 7 Edition continues reading the JCL, the following statement is encountered:

```
//   NMDPROCL,PARM1=data,PARM2=data
```

CA WA CA 7 Edition looks for the member NMDPROCL in the local DPROC library (HLQ.DPROC.LOCAL) and finds the member, so CA WA CA 7 Edition resolves the JCL. CA WA CA 7 Edition does not look in the CARPROC DD because it already found the member.

Usage notes:

- To allocate a local DPROC library, code DPROC= on the JCL statement in the initialization file. For information about coding DPROC, see the *CA WA CA 7 Edition Systems Programming Guide.*
- If JCL definitions are stored in the CA WA CA 7 Edition VRM data set, associate a local DPROC library using the /JCL command. For information about the /JCL command, see the CA WA CA 7 Edition *Command Reference Guide*.

Sample Parameter Data Set

Example: Create a Parameter Data Set

This example shows CA WA CA 7 Edition parameters in a partitioned data set member that is used when specifying parameters using EDCHEK ISPF:

```
EDIT       USERID1.JCK.CNTL(RESPARMS) - 01.21            Columns 00001 00072
Command ===>                                               Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 * THESE ARE CA JCLCHECK - CA 7 INTERFACE PARMS FOR JOB, PAYROLL9.
000002 SOURCE=E
000003 W_JCLLIB=&WORKJCL
000004 MEMBER=PAYROLL9
000005 DATE=11353
000006 TIME=2000
000007 SCHID=1
000008 NODE=A31SENF
000009 INSTANCE=C377
****** **************************** Bottom of Data ****************************
```

CA WA ESP Edition Parameters

This topic contains the CA WA ESP Edition parameters and the following example:

- Sample Parameter Data Set (see page 104)

This procedure describes how to specify the parameters.

Follow these steps:

1. Specify the CA WA ESP Edition parameters in a sequential file (including partitioned data set member).
2. Start the parameters in column 1 and stop on or before column 71.
3. Specify the parameters in any order (there is no correct parameter order).
4. Specify comment lines (if necessary) by placing an asterisk (*) in column 1. Do not add comments to the right side of the parameter value.

The following parameters are for CA WA ESP Edition:

COMMAND=*command*

Identifies the CA WA ESP Edition command to invoke. This parameter is mandatory and can only be set to SIMULATE or SCAN.

Example: COMMAND=SIMULATE

SUBSYS=*subsys*

Identifies the CA WA ESP Edition subsystem name. If no subsystem name is specified, ESP is used.

Example: SUBSYS=ESPP

SYMBOL=*symbol*

Identifies the symbol-introducer character. If no symbol is specified, the character is taken from the subsystem.

Example: SYMBOL=%

EVENT=*eventid*

Identifies the name of the event for which CA WA ESP Edition simulates processing. This field is mandatory when invoking the SIMULATE command.

Example: EVENT=CYBER.PAYROLL

SCHED=*schedtime*

Represents the schedule time for the job. The time can be any valid time specification on a SCHEDULE statement. When *schedtime* contains separator characters (commas or blanks), it must be enclosed in quotes. If no schedule time is specified, the default is the next scheduled time.

Example: SCHED=TOMORROW

SYMLIB=*symlib*

SYMLIB2=*symlib*

SYMLIB3=*symlib*

Identifies the names of the data sets to be used as symbol libraries for symbolic variable generation. The parameters SYMLIB2 and SYMLIB3 can be used when the data set names do not fit on the SYMLIB line. All file names should be fully-qualified.

Example: SYMLIB=USERID1.JCKESP.CNTL(SYMLIB)

CALENDAR=*cal*

CALENDR2=*cal*

CALENDR3=*cal*

Identifies the calendars that are retrieved for this SCAN. The parameters CALENDR2 and CALENDR3 can be used when the calendar names do not fit on the CALENDAR line.

Example: CALENDAR=CAL2011

PROCJOB=*jobname*

Identifies the fully qualified CA WA ESP Edition job name, which endures process-mode simulation. This parameter is a mandatory field when invoking the SIMULATE command in ISPF. If PROCJOB is not specified in batch, the job name that is used comes from the JOBNAME parameter (specified in the RESPARM parameter). If PROCJOB is not specified when running under EDCHEK, the job name that is used comes from the ISPF EDIT session member name.

Example: PROCJOB=C.WICKEDWITCH

VARS=*var_list*

VARS2=*var_list*

VARS3=*var_list*

VARS4=*var_list*

VARS5=*var_list*

Provides a list of variable names and values. These variables can include monitor and signal variables. The parameters VARS2, VARS3, VARS4, and VARS5 can be used when the variables do not fit on the VARS line.

Example: VARS=MNJOB(PAYJOB1),MNAPPL(PAYROLL)

USER1=*user_data*

USER12=*user_data*

Identifies the USER1 variable passed to the event. The parameter USER12 can be used when the variable does not fit on the USER1 line.

Example: USER1=IEFBR14

USER2=*user_data*

USER22=*user_data*

Identifies the USER2 variable passed to the event. The parameter USER22 can be used when the variable does not fit on the USER2 line.

Example: USER2=SYSIN

USER3=*user_data*

USER32=*user_data*

Identifies the USER3 variable passed to the event. The parameter USER32 can be used when the variable does not fit on the USER3 line.

Example: USER3=SYSPRINT

USER4=*user_data*

USER42=*user_data*

Identifies the USER4 variable passed to the event. The parameter USER42 can be used when the variable does not fit on the USER4 line.

Example: USER4=SYSUDUMP

JCLSCAN=exit_rtn

Identifies the name of the JCLSCAN exit module.

Example: JCLSCAN=CYBJCSCN

RECORD

Informs CA JCLCheck to record the resolution session. The record data is written to the partitioned data set allocated in the RECORD DD statement. Allocate this file as a PDS whose DCB is LRECL=500, RECFM=FB, and DSORG=PO. Under EDCHEK and ISPF, the name of this file must be specified in the JCKUSRES CLIST. In a batch job, the JCL deck must contain a RECORD DD statement.

Use this parameter only under the direction of CA Support.

Example: RECORD

RECORDSEQ

Informs CA JCLCheck to record the resolution session. The record data is written to a sequential data set allocated in the RECORDSQ DD statement. Allocate this file as a sequential file whose DCB is LRECL=500, RECFM=FB. Under EDCHEK and ISPF, the name of this file must be specified in the JCKUSRES CLIST. In a batch job, the JCL deck must contain a RECORDSQ DD statement.

Use this parameter only under the direction of CA Support.

Example: RECORDSEQ

Sample Parameter Data Set

Example: Create a Parameter Data Set

This example shows CA WA ESP Edition parameters in a partitioned data set member that is used when specifying parameters using EDCHEK ISPF:

```
EDIT       USERID1.JCK.CNTL(RESPARMS) - 01.21            Columns 00001 00072
Command ===>                                               Scroll ===> CSR
****** ***************************** Top of Data ******************************
000100 RECORD
000200 RECORDSEQ
000300 SUBSYS=ES51
000400 COMMAND=SIMULATE
000500 EVENT=USERID1.APPL1
000600 PROCJOB=IDCAMS.TEST
000700 SYMLIB=USERID1.JCK.CNTL(SYMLIB)
000800 USER1=IEFBR14
001000 *USER3=IDCAMS
001100 JCLSCAN=CYBJCSCN
********************************** Bottom of Data ***************************
```

Control-M Parameters

This topic contains the Control-M parameters and the following example:

- Sample Parameter Data Set (see page 107)

This procedure describes how to specify the parameters.

Follow these steps:

1. Specify the Control-M parameters in a sequential file (including partitioned data set member).
2. Start the parameters in column 1 and stop on or before column 71.
3. Specify the parameters in any order (there is no correct parameter order).
4. Specify comment lines (if necessary) by placing an asterisk (*) in column 1. Do not add comments to the right side of the parameter value.

For Control-M, two types of statements can be specified in the Resolve Parameters:

1. RESOLVE Parameters—required for resolve processing
2. CTMAESIM Options—passed to the CTMAES Program

The following parameters are the RESOLVE parameters:

RECORD

Informs CA JCLCheck to record the resolution session. The record data is written to the partitioned data set allocated in the RECORD DD statement. Allocate this file as a PDS whose DCB is LRECL=500, RECFM=FB, and DSORG=PO. Under EDCHEK and ISPF, the name of this file can be specified in the JCLUSRCM CLIST. In a batch job, the JCL deck must contain a RECORD DD statement.

Use this parameter only under the direction of CA Support.

Example: RECORD

RECORDSEQ

Informs CA JCLCheck to record the resolution session. The record data is written to a sequential data set allocated in the RECORDSQ DD statement. Allocate this file as a sequential file whose DCB is LRECL=500 and RECFM=FB. Under EDCHEK, a RECORDSQ DD statement must be allocated before calling CA JCLCheck. In a batch job, the JCL deck must contain a RECORDSQ DD statement. This option is not supported under the ISPF interface.

Use this parameter only under the direction of CA Support.

Example: RECORDSEQ

SOURCE=I|E

Identifies the source of the input JCL.

I—Internal—The JCL source is specified using CTMAESIM options (that is, LIBRARY/MEMBER or SCHDLIB/TABLE/JOB) in the resolve parameter file. CTMAESIM processes these options directly. This parameter is the default.

E—External—The input JCL comes from an external file that a SYSIN DD statement defines. The SYSIN must be a sequential file. Any CTMAESIM options that reference JCL that are found in the resolve parameter file are ignored.

Example: SOURCE=E

Use this parameter only in Batch.

ERES

Simulates EDCHK processing in batch and is intended for use only by CA Support. This parameter requires specially modified input JCL that is provided on the SYSIN DD statement. This parameter also requires the member name for the job that is specified on the JOBNAME() parameter on the EXEC statement.

Use this parameter only under the direction of CA Support.

Example: ERES

Use this parameter only in Batch.

JCLPRFX=

When SOURCE=E is used or the CTM Resolve option is used in EDCHEK (EJCK), a file is created to hold the JCL member. This file is deleted after CTMAESIM is called. The allocated file is named *jclprfx*.JCKTEMP. By default, the prefix *jclprfx* is the USER ID of the current user. The JCLPRFX option allows for an alternative file prefix to be specified. You can specify up to 17 characters.

Example: JCLPRFX=WORK.TEST

CTMPRINT

When the CTMAESIM program is called, a report is returned. By default, this report is examined for error messages that can be reported. This report includes, in addition to error messages, the complete resolved JCL for the job. By default, only error messages are returned for the job. Include this option if you would like to see the entire report.

Example: CTMPRINT

The following bullets describe the CTMAESIM options:

- CTMAESIM has options for selecting JCL needing resolution, and for setting parameters that can affect the parameter resolution.
- CTMAESIM allows JCL selection in two modes: Library and Scheduling. Some options are only used with a particular mode and other options that can be used with either mode. Common CTMAESIM parameters are discussed following this bullet. For a complete list of parameters available, see the section titled 'CTMAESIM — Test AutoEdit Syntax' in the *CONTROL-M Utilities Guide*.
- CA JCLCheck permits any valid CTMAESIM option in the RESPARMS file.
- The resolve option under EDCHEK (EJCK) only supports Library mode and provides the needed LIBRARY, MEMBER, and SUBMIT command options. Any Library mode or general option can be specified, except for LIBRARY, MEMBER, and the commands (SUBMIT, LIST, ...) that are provided. EDCHEK generates these statements to define the location of a temporary file containing the JCL that is being edited. If these restricted parameters are specified, they are ignored.
- In the CA JCLCheck ISPF panels, all valid CTMAESIM options can be specified in Library or Scheduling mode. Always specify the command field as SUBMIT. When any of the other command field values are specified CA JCLCheck changes these values to SUBMIT before calling the CTMAES utility. For example, LIST, SEND, or SUBSCAN.
- In a batch job, the RESPARM requirements depend on the setting of the SOURCE= parameter. When SOURCE=Internal is specified, then the same requirements that are described for the CA JCLCheck ISPF panels apply. When SOURCE=External or RESOLVE(CTM,NOPARM) is specified, then the same requirements that are described for EDCHEK apply. Note the added requirement that the SYSIN DD point to a sequential file containing the JCL to process.

Sample Parameter Data Set

Example: Create a Parameter Data Set

This example shows Control-M parameters in a partitioned data set member that is used when specifying parameters using EDCHEK ISPF:

```
EDCHEK RESPARMS Sample
EDIT       USERID1.JCL.CNTL(RESPARMS) - 01.26         Columns 00001 00072
Command ===>                                            Scroll ===> CSR
****** ************************** Top of Data ***************************
000002 OWNER SY01
000003 WDATE 060600
000004 ODATE 060600
****** ************************* Bottom of Data ************************
```

This example shows Control-M parameters in a partitioned data set member that is used when specifying parameters using ISPF:

```
ISPF RESPARMS Sample
EDIT       USERID1.SPF.RESPARMS                  Columns 00001 00072
Command ===>                                       Scroll ===> PAGE
****** ************************* Top of Data ************************
000130 OWNER SY01
000140 WDATE 060600
000150 ODATE 060600
000200 LIBRARY TEST.JCL.CNTL
000300 MEMBER CTMJ*
000400 SUBMIT
000600 MEMBER CTMT*
000700 SUBMIT
****** ************************ Bottom of Data ************************
```

Tivoli Workload Scheduler Parameters

This topic contains the TWS parameters and the following example:

- Sample Parameter Data Set (see page 112)

This procedure describes how to specify the parameters.

Follow these steps:

1. Specify the TWS parameters in a sequential file (including partitioned data set member).
2. Start the parameters in column 1 and stop on or before column 71.
3. Specify the parameters in any order (there is no correct parameter order).
4. Specify comment lines (if necessary) by placing an asterisk (*) in column 1. Do not add comments to the right side of the parameter value.

The following parameters are for TWS:

SUBSYS=*subsys*

Identifies the TWS subsystem name. If no TWS subsystem name is provided, OPCC is used. This parameter must be present on the first non-comment line.

Example: SUBSYS=OPOP

ADID=*applidname*

Identifies the 16-character application ID that is used for selection in batch and for variable substitution in JCL in EDCHEK. This parameter is mandatory.

Example: ADID=DFAG0D00BCV2GRCA

IA=*yymmddhhmm*

Specifies the ten-character input arrival date and time that is used for selection in batch and for variable table selection in EDCHEK. This field defaults to the current date and time.

Example: IA=1004301800

W_ADID=*applidname*

Identifies the 16-character work application ID. This application ID is created to hold external JCL for resolution temporarily. You can use W_ADIDC and W_ADIDT instead to create the work ADID. This field defaults to JCLCHECKDEFAULT.

Example: W_ADID=DFAG0D00BCV2GRCA

W_ADIDC=*nn*

Specifies a two character numeric value between 01-15.

W_ADIDT=*xx*

When the W_ADID keyword is omitted or set to "JCLCHECKDEFAULT", the work ADID (W_ADID) value can be created from the production ADID specified in OADID. The value in OADID is modified starting at the column specified in W_ADIDC. W_ADIDT specifies the two characters to substitute. W_ADIDC must be set to a two-character numeric value between 01-15. W_ADIDT must be set to a two-character text value.

Example: W_ADIDC=09 W_ADIDT=PR

If OADID is set to PRODAPPLTX023, the work application name (W_ADID) that would be generated is PRODAPPLPR023.

VALID=*yymmdd*

Specifies the date on which the application is valid. This field defaults to the current date.

Example: VALID=100722

W_WSNAME=*wsname*

Specifies the four-character workstation name that is used for operations in the work application. This workstation must be a valid Type C (computer) workstation that is defined to TWS. This parameter is mandatory if SRCETYPE=SEQ.

Example: W_WSNAME=WSPR

W_OPNO=*opno*

Specifies the three-digit work operation number (0-255). If any TWS variables in the JCL refer to &OOPNO, this value is substituted. This parameter is mandatory if SRCETYPE=SEQ.

Example: W_OPNO=10

W_GROUP=*groupid*

Specifies the eight-character group definition ID that is used for the work application.

Example: W_GROUP=GROUPPRD

W_OWNER=*ownerid*

Specifies the 16-character owner ID that is used for the work application.

Example: W_OWNER=USERID1

ADRUN=YES|NO

By default, CA JCLCheck uses the ADRUN option on the INSERT APPLICATION to prevent TWS from submitting the application for execution. Sites that use the TWS GROUPID option to specify run dates must specify ADRUN=NO as TWS does not allow both. Care must be taken when using ADRUN=NO to insure TWS does not select the application for submission. When using the TWS interface through the ISPF panels, specify the ADRUN option in the Appl Run Dates (Y or N) field (panel JCK0207).

&OADID=&T*cccc*

Specifies a six-character temporary variable name that is used internally to replace the &OADID variable in the JCL during the variable resolution process. In TWS, the &OADID variable can be used in JCL to specify the Application ID. Since CA JCLCheck uses a work application, the value that is supplied by TWS in the &OADID variable is the work application ID (not the original application ID). The original application ID can be inappropriate for the JCL. This temporary variable name is used internally instead of the &OADID variable in the JCL. The name must begin with “&T”, then four alpha-numeric characters that equal a total of six characters in length. Also, do not match this name with any other variable name that is defined in the JCL. If not specified, &TJCG0 is used.

Example: &OADID=&TJCHK

SIMTIME=*yyyymmddhhmm*

Specifies the 12-character simulated date and time. The *yyyy* can be any year between 1984 - 2071. This field defaults to the current date and time.

Example: SIMTIME=201007011248

JCLVTAB=*table*

Identifies the 16-character JCL variable table name for the work application. In a batch job, it defaults to the value specified in the application. In ISPF, if not specified, the default is the table specified in the application or the system default.

Example: JCLVTAB=TESTVARIABLETBL

SOURCE=I|E|EI

Identifies the source of the input JCL.

I—Internal—The input JCL is pulled directly from TWS.

E—External—The input JCL comes from an external file that a SYSIN DD statement defines.

EI—External/Internal—The input JCL comes from an external file that a SYSIN DD statement defines. However, when a particular member is not found in the SYSIN file, the internal JCL for the event is used.

Example: SOURCE=E

SRCETYPE=SEQ|PDS|LIB|PAN

Identifies the format of the SYSIN file when SOURCE=E or EI. This parameter is mandatory when SOURCE=E or EI. This parameter is ignored if SOURCE=I.

Example: SRCETYPE=SEQ

SNAP

Informs CA JCLCheck to take diagnostic snap dumps. The dumps are written to the file allocated in the SNAP DD statement. Allocate this file as a SYSOUT file or a file whose DCB is LRECL=125, BLKSIZE=882, RECFM=VBA, and DSORG=PS. Under EDCHEK, a SNAP DD statement must be allocated before calling CA JCLCheck. Under the ISPF interface, a data set with the name of the Resolve Parm File with an '.S' appended to the end of it is allocated automatically. In a batch job, the JCL deck must contain a SNAP DD statement.

Use this parameter only under the direction of CA Support.

Example: SNAP

RECORD

Informs CA JCLCheck to record the resolution session. The record data is written to the partitioned data set allocated in the RECORD DD statement. Allocate this file as a PDS whose DCB is LRECL=500, RECFM=FB, and DSORG=PO. Under EDCHEK, a RECORD DD statement must be allocated before calling CA JCLCheck. Under the ISPF interface, a data set with the name of the Resolve Parm File with an '.R' appended to the end of it is allocated automatically. In a batch job, the JCL deck must contain a RECORD DD statement.

Use this parameter only under the direction of CA Support.

Example: RECORD

RECORDSEQ

Informs CA JCLCheck to record the resolution session. The record data is written to a sequential data set allocated in the RECORDSQ DD statement. Allocate this file as a sequential file whose DCB is LRECL=500 and RECFM=FB. Under EDCHEK, a RECORDSQ DD statement must be allocated before calling CA JCLCheck. In a batch job, the JCL deck must contain a RECORDSQ DD statement. This option is not supported under the ISPF interface.

Use this parameter only under the direction of CA Support.

Example: RECORDSEQ

Note: The ADID and IA values are used to select the jobs that are processed by accessing the application records in TWS. The ADID and IA values are also used to retrieve the list of operations (jobs) and dependent operations in the application. These values provide the job names and operation numbers for each job being processed, as well as the execution order for the operations. The execution order is the order CA JCLCheck processes the operations.

Sample Parameter Data Set

Example: Create a Parameter Data Set

This example shows TWS parameters in a partitioned data set member that is used when specifying parameters using EDCHEK ISPF:

```
EDIT       USERID1.JCK.CNTL(RESPARMS) - 01.21            Columns 00001 00072
Command ===>                                               Scroll ===> CSR
****** ***************************** Top of Data ******************************
000100 * The following are TWS parameters
000200 SUBSYS=OPOP
000300 SOURCE=E
000400 SRCETYPE=SEQ
000500 ADID=TWSAPPLICATION
000600 IA=1004301800
000700 W_ADID=JCLCHECKDEFAULT
000800 W_WSNAME=WSPR
000900 W_OPNO=10
001000 &OADID=&TJCG0
001100 SIMTIME=201007300810
****** **************************** Bottom of Data ****************************
```

Executing JCL DD Statements

Use the following EXEC statement to invoke CA JCLCheck:

EXEC

Calls the JCLCheck program and supplies all the PARM requirements. For a complete description of all the PARM options available, see Setting up CA JCLCheck Options (see page 18).

This EXEC statement is *required*.

STEPLIB

Statement that points to the JCLCheck load library to call the JCLCheck procedure.

The STEPLIB statement is only required if it is not set up in the linklist.

Use the following list of required and optional DD statements when you invoke CA JCLCheck:

SYSPRINT

File that receives the CA JCLCheck reports. Its logical record length is 133 and it can be blocked. To change the ddname of this file, use the LIST option.

Required unless PARM=NOLIST is specified or the NOLIST option is specified in the options file.

SYSPROC

Statement that points to the procedure library that CA JCLCheck uses to resolve references to cataloged procedures. If you have more than one procedure library, you can concatenate other data sets on DD statements to this one. To change the ddname of this file, use the PROC option. Specify the same procedure libraries for this file as you have specified in your JES procedure. Generally, the ddname is PROC00 for JES2 and IATPLBST for JES3.

Required unless you are using AUTOPROC.

CAIVMFI

CA TLMS Volume Master File (VMF).

Required if CA TLMS support is installed and you select the CA TLMS (TLMS) support option.

PROC*nn*

Provides CA JCLCheck support for the PROCLIB parameter of the JES2 /*JOBPARM control statement.

Required unless you are using AUTOPROC. You should add the PROCLIB allocation statements from the JES2 procedure to the JCL that executes CA JCLCheck.

Do not use with AUTOPROC.

IATPLB*nn*

Provides CA JCLCheck support for the PROC= parameter of the JES3 //*MAIN control card. If you desire support, add the alternate PROCLIB allocation statements from your JES3 procedure to the JCL which executes CA JCLCheck.

Optional.

JCHKFREE

Marks the beginning of the sequence of DD statements that describe the data sets that you want CA JCLCheck to dequeue.

Optional unless you specify PARM=DEQUE.

JENDFREE

Specifies the end of the sequence of DD statements that you want CA JCLCheck to free.

Optional unless you specify PARM=DEQUE.

SYSGRAPH

Receives the Flow Diagram Report (Report #10). The default ddname is SYSGRAPH. Specify another ddname by using the GRAPH option. For information about the GRAPH option, see the *Command Reference Guide*. The default and minimum possible value for LRECL is 133.

Optional unless you specify PARM=GRAPH.

COBLIB

Use with the COBEXIT option. The COBLIB statement describes a load library that can contain a CA JCLCheck COBOL exit routine. When you include this statement, CA JCLCheck loads the COBOL exit program named in the COBEXIT option from this library. Use this statement to facilitate development and testing of COBOL exit programs. For information about COBOL exits, see the P*rogramming Guide.*

Optional.

SYSDICT

Accesses the CA APCDOC Cross-Reference database. The SYSDICT DD statement specifies the cluster data set name for this database.

Required only if you install the CA APCDOC interface.

SYSDICT1

Accesses the CA APCDOC Cross-Reference database. The DD statement specifies the path name for the Cross-Reference database.

Required only if you install the CA APCDOC interface.

SYSTERM

Contains the terminal report file. The logical record length is 80 and can be blocked. You can change the ddname of this file with the TERM option.

Note: For information about the TERM option, see the *Command Reference Guide*.

Optional unless you specify PARM=TERM.

SYSRDR

Defines the destination for error-free job streams. JES2 automatically processes these jobs if the statement is coded as shown on the previous page. CA JCLCheck writes the JCL for jobs containing errors to the point of the error, and then terminates with a JES2 /*PURGE (or HASP /*DEL) control statement.

Optional unless you specify PARM=SUBMIT.

SYSDD*xxx*

In non-MVS systems, needed for the dynamic allocation of data sets required by the CTLSCAN, EASYPROC, or PXREF options. When using any of these options, you should add a SYSDD*xxx* DD statement for every volume that contains a control statement data set (for CTLSCAN), a private procedure library (for EASYPROC), or a program library (PXREF). In practice, it is probably best to include a volume allocation DD statement for every resident or reserved volume in the installation, as part of a standard CA JCLCheck procedure. You can change the ddname prefix on these DD statements with the PREFIX option.

Optional in non-MVS and not used in MVS.

SYSIN

Points to the JCL stream that you want CA JCLCheck to analyze. CA JCLCheck also requires the DLM parameter if an input stream data set contains any statements beginning with "/*" (such as JES2 control statements). This data set is either a card image file, a partitioned data set (with the PDS option), a CA Panvalet library (with the Panvalet option), or a CA Librarian library (with the Librarian option). Change the ddname of this file with the Input option.

Note: For more information, see the runtime options in the *Command Reference Guide*.

SYSIN requires an LRECL of 80.

Required.

STDRULE

Accesses the CA JCLCheck Standards Rules database. This DD statement specifies the data set name for the database.

Optional.

SYSTSPRT

Use with the STDREXX option. Receives the REXX error messages and output from REXX commands. Its logical record length is 133, and it can be blocked. The ddname may be different if your site has changed the REXX installation defaults.

Optional.

SYSEXEC

Points to the REXX EXEC library used with the STDREXX option.

Optional.

Chapter 3: How to Use Job Control Standards

Job Control Standards (JCS) is an ISPF interface that allows you to specify your site's JCL standards for compliance or enforcement checking by CA JCLCheck. These standards are maintained by a Standards Administrator and give the administrator the ability to set the conditions to check and specify error messages associated with those conditions.

Job standards can also be enforced using REXX. The REXX support allows for more flexibility in checking and enforcing standards. CA Technologies recommend using REXX instead of JCS for standards enforcement.

The following examples are set up in a step-by-step format to show you how to use the Job Control Standards panels.

Note: For information about installing Job Control Standards, see the *Installation Guide*. For information about Job Control Standards panels, see the *Programming Guide*.

This section contains the following topics:

Masks (see page 118)
Adding a Standard (see page 118)
Adding a Standard with More Than One Rule (see page 126)
Deleting a Standard (see page 136)
Deleting a Rule Related to an Error Message (see page 137)
Deleting Action Rules from a Standard (see page 140)

Masks

The Standards Selection and Standards Rule Definition panels allow you to use masks (also called *generics* or *wildcards*). Masks allow you to match one or more selected characters. A mask is an asterisk (*) for matching a string of any number of characters or a question mark (?) for matching any single character (only one). For example:

- Specifying TEST*1 displays all items beginning with TEST and ending with 1 (for example, TESTDIR1, TESTD1, TESTPRD1, TESTP1).
- Specifying TEST* displays all items beginning with TEST (for example, TESTDATA, TESTDIR1, TESTD1, TESTPRO, TESTPROD, TESTPRD1, TESTP1).
- Specifying TEST?1 displays all six-character names beginning with TEST and ending with 1 (for example, TESTD1, TESTP1).

In some cases, using masks results in the matching of more than one name. Whenever this happens in CA JCLCheck, a directory of all the names that match the user-entered name is displayed, and you are asked to choose from the directory.

Adding a Standard

In this exercise you create a standard called TESTSTD on the Standard Rules database. This standard directs CA JCLCheck to check all test jobs for job names that begin with T. If the job name does not begin with T, it directs CA JCLCheck to display an error message.

To do this, you must access a series of panels and perform a number of steps to define the criteria for the TESTSTD standards rule and the action that is the result of this rule.

The following are the basic steps for adding a standard:

1. Create the standard name
2. Select the statement and parameter type
3. Define the standards rule
4. Select the action
5. Define the action
6. Confirm the standard
7. Compile the standard
8. Use the standard
9. Print the standard

To access the Job Control Standard panels, enter **%JCKSTD** at the TSO READY prompt or as a TSO command at the OPTION line if you are in ISPF.

Creating the Standard Name

Use this panel to create the new standard TESTSTD.

1. Scroll forward (PF8) until you see the JOB statement types and enter **E** in the OPTION field.
2. Enter **TESTSTD** in the Standard Group Name field.

   Press Enter to move on to the next panel.

```
CAZ1RPSP -----------CA JCLCheck: STANDARDS (SELECTION)  --------------
OPTION ===> E

blank         - Display Standard directory
E EDIT        - Edit/Create a Standard
D DELETE      - Delete a Standard
C COPY        - Copy a Standard
R RENAME      - Rename a Standard

Masks
  * - Match any String
  ? - Match any Character

Standard Group Name ==> TESTSTD
```

Selecting the Statement and Parameter Type

The STANDARDS UPDATE panel (CAZ1RTSP) displays the allowable statement and parameter types for which you can define a JCL standard. You use this panel to begin defining a rule for TESTSTD that relates to the JOB NAME statement and parameter type.

To select the statement and parameter type, enter **S** in the CMD field preceding JOB NAME, and press Enter.

```
CAZ1RTSP ---------  CA JCLCheck: STANDARDS (UPDATE)  ----------  ROW 46 OF 61
COMMAND ===>                                                  SCROLL ===> PAGE

      STANDARD GROUP NAME :    TESTSTD

  To select a Statement Type :  Enter 'S' in CMD field, press Enter
  To return to previous Panel :  Press End
  CMD  TYPE     PARAMETER       SUB-PARAMETER
       EXEC     TIME_2          SECONDS
       JOB      ACCOUNT         NUMBER
       JOB      ADDRSPC
       JOB      CLASS
       JOB      GROUP
       JOB      MSGCLASS
   S   JOB      NAME
       JOB      NOTIFY
       JOB      PERFORM
       JOB      PGMR
       JOB      PRTY
       JOB      REGION
       JOB      TIME_1          MINUTES
       JOB      TIME_2          SECONDS
       JOB      USERID
```

Defining the Standards Rule

The STANDARDS RULE DEFINITION panel (CAZ1R01P) is an edit panel on which you define rules consisting of a relational operator, a value for that operator, and an action name. The action name, its operator, and value result in a specific action, which is initiated during standards checking.

You use this panel to continue defining the rule for standard TESTSTD to direct CA JCLCheck to produce an action named ERR100 if the job name is not equal to T.

1. Enter **NE (not equal to)** in the OPER field.
2. Enter **T*** in the Value field.
3. Enter **ERR100** in the Action field. (You can name the action anything you like, in accordance with your company's standards.)

   Press End when you finish filling in this panel to move on to the next step.

```
CAZ1R01P ------- CA JCLCheck: STANDARDS (RULE DEFINITION) ------- ROW 1 OF 1
COMMAND ===>                                                SCROLL ===> PAGE

  STANDARD GROUP NAME   :  TESTSTD       CMD Field Usage:
  STATEMENT TYPE        :  JOB            R - Replicate Line, Press Enter
  PARAMETER TYPE        :  NAME           I - Insert Line, Press Enter
  SUB PARAMETER TYPE    :                 D - Delete Line, Press Enter
  ACTION NAME           :  INITIAL        S - Select Existing Action, Press Enter
       Press End Returns to Previous Panel

  CMD  OPER     VALUE    ACTION
        NE        T*     ERR100
 ****************************** BOTTOM OF DATA ************************
```

Selecting the Action

The RELATED STANDARDS (UPDATE) panel (CAZ1RXSP) lets you select the statement and parameter type that you need to use to produce the action taken on this rule.

In this case, select ERROR DEFINITION to begin creating an error message for this rule.

- Scroll forward (PF8) until you see the ERROR statement type and enter **S** in the CMD field preceding the ERROR DEFINITION statement and parameter type, and press Enter.

```
CAZ1RXSP -----  CA JCLCheck: RELATED STANDARDS (UPDATE)  -------  ROW 1 OF 63
COMMAND ===>                                                SCROLL ===> PAGE

  To select a Statement Type  :  Enter 'S' in CMD field, press Enter
  To return to previous Panel :  Press End

       STANDARD GROUP NAME : TESTSTD            ACTION : ERR100

  CMD  TYPE     PARAMETER       SUB-PARAMETER
       ERROR    DEFINITION      FAST PATH
       JOB      NAME
       =======  ============    ================
       DD       AVGREC
       DD       COPIES
       DD       DATACLAS
```

```
CAZ1RXSP -----  CA JCLCheck: RELATED STANDARDS (UPDATE) ------  ROW 31 OF 63
COMMAND ===>                                                  SCROLL ===> PAGE

   To select a Statement Type  :  Enter 'S' in CMD field, press Enter
   To return to previous panel :  Press End

       STANDARD GROUP NAME : TESTSTD            ACTION:  ERR100

       CMD      TYPE     PARAMETER        SUB-PARAMETER
                ERROR    DEFINITION       FAST PATH
                DD       SYSOUT           CLASS
                DD       SYSOUT_2         WRITER
                DD       SYSOUT_3         FORM
                DD       UCS_1            CODE
                DD       UNIT
                DD       VOLUME_5         SERIAL_#
        S       ERROR    DEFINITION
                EXEC     ACCOUNT          INFO
```

Defining the Action

Use the STANDARDS DEFINITION panel (CAZ1RERP) to define error messages. The information entered here is the message that you see whenever this error occurs.

To define the messages for ERR100:

1. Enter **E** in the Severity field.
2. Enter **100** in the Message# field.
3. Enter ALL TEST JOBS MUST HAVE A JOB NAME BEGINNING WITH T in the Text field, and press End.

   Note: When you press End, CA JCLCheck backs out through all previous Rule Definition panels. Continue to press End until the STANDARDS CONFIRMATION panel appears.

```
CAZ1RERP -------  CA JCLCheck: STANDARDS (DEFINITION)  -----------------
COMMAND ===>                                                  SCROLL ===> PAGE


      Delete Indicator:
      To delete Error enter D and press End

    STANDARD GROUP NAME : TESTSTD
    STATEMENT TYPE      : ERROR
    PARAMETER TYPE      : DEFINITION
    ACTION NAME         : ERR100

    SEVERITY    : E   ( I - INFORMATIONAL, W - WARNING, E - ERROR, S - SEVERE)
    MESSAGE#    : 100
    TEXT        : ALL TEST JOBS MUST HAVE A JOB NAME BEGINNING WITH T
```

Confirming the Standard

The STANDARDS CONFIRMATION panel (CAZ1RUCP) adds the standard to the Standard Rules database or cancels your request.

1. Press End to add TESTSTD to the Standard Rules database.
2. Press Cancel to cancel the request.

```
CAZ1RUCP -------- CA JCLCheck: STANDARDS (CONFIRMATION) ------------------
COMMAND ===>                                              SCROLL  ===> PAGE


         Standard Name:  TESTSTD


    Press End to update the above standard on the rule database
    Use   CANCEL command to cancel the request
```

Compiling the Standard

To compile the standard TESTSTD so that CA JCLCheck can use it, execute CAZ2JCL member Z1JCSCMP, which consists of the following steps:

- EXEC PROC CAZ1JCS

  This step executes the CAZ1EVAL program, which reads the Rules Database, evaluates all the standards, and creates source code for a standards exit. The source code is named CAZ1SUSR and is placed in the library specified by JCL symbolic OPTLIB.

- EXEC PROC CAZ1ASM

  This step assembles CAZ1SUSR.

- EXEC PROC CAZ1LNK

  This step links CAZ1SUSR and places it in CAILOAD.

Note: You must perform this compile whenever additions or changes are made to any standards definitions, as they all assemble and link into the load module CAZ1SUSR. The name of this load module *must not be changed.*

Using the Standard

Once Step 7 is complete, use the standard by specifying the runtime option STANDARD (TESTSTD) when you execute CA JCLCheck to perform site-dependent validation for test job JCL.

Here is an example of how you invoke the standard using EDCHEK:

```
 ------------------------------------------------------------------------
 COMMAND ===> EJCK STANDARD(TESTSTD)                     SCROLL ==> PAGE
****************************** TOP OF DATA ******************************
000001 //PRODJOB JOB (40100000),'USER001',USER=USER001,
000002 //      MSGCLASS=X,CLASS=K
000003 //*
000004 //CAZ1RLST EXEC PGM=CAZ1RLST,PARM='A@2'
000005 //STEPLIB  DD  DSN=LESAN01.TEST.LOAD,DISP=SHR
000006 //STDRULE  DD  DISP=SHR,
000007 //             DSN=CACT.DEVL.CAJ6.STANDARD
000008 //SYSOUT   DD  SYSOUT=*
000009 //SYSPRINT DD  SYSOUT=*

************************** BOTTOM OF DATA ******************************
```

After using the TESTSTD standard, EDCHEK produces the following sample error message:

```
 ------------------------------------------------------------------------
 COMMAND ===>                                            SCROLL ===> PAGE
************************************ TOP OF DATA ************************
==MSG> CAY6000 1 STATEMENT FLAGGED IN JOB "PRODJOB" MAXIMUM SEVERITY WAS 8
==MSG>
==MSG>
==MSG>
000001 //PRODJOB JOB (40100000),'USER001',USER=USER001'
==MSG> //PRODJOB JOB (40100000),"USER001",USER=USER001'
000002 //       MSGCLASS=X,CLASS=K
==MSG> //       MSGCLASS=X,CLASS=K
==MSG> CAY6341 "100 -- ALL TEST JOBS MUST HAVE A JOB NAME BEGINNING WITH T"
==MSG>
==MSG> CAY6340 JOB CONTROL STANDARDS VIOLATION DETECTED
==MSG>
000003 //*
000004 //CAZ1RLST EXEC PGM=CAZ1RLST,PARM='A@2'
000005 //STEPLIB  DD  DSN=LESAN01.TEST.LOAD,DISP=SHR
000006 //STDRULE  DD  DISP=SHR
000007 //             DSN=CACT.DEVL.CAJ6.STANDARD
000008 //SYSOUT       DD  SYSOUT=*
000009 //SYSPRINT DD  SYSOUT=*

****************************** BOTTOM OF DATA ***************************
```

Note: All standards violations use message number CAY6341. The message number you defined (in this case, 100) is placed at the beginning of the message text.

This is a sample of an EDCHEK execution where no errors occur. In other words, the JCL conforms to the site-dependent standards.

```
 ------------------------------------------------------------------------
 COMMAND ===> EJCK STANDARD(TESTSTD)                    SCROLL ==> PAGE
******************************** TOP OF DATA ************************
==MSG> CAY6000 NO STATEMENTS FLAGGED IN JOB "TESTJOB" MAXIMUM SEVERITY WAS 0
==MSG>
==MSG>
==MSG>
000001 //TESTJOB JOB (40100000),'USER001',USER=USER001,
000002 //         MSGCLASS=X,CLASS=K
000003 //*
000004 //CAZ1RLST EXEC PGM=CAZ1RLST,PARM=A@2'
000005 //STEPLIB  DD  DSN=LESAN01.TEST.LOAD,DISP=SHR
000006 //STDRULE  DD  DISP=SHR
000007 //             DSN=CACT.DEVL.CAJ6.STANDARD
000008 //SYSOUT   DD  SYSOUT=*
000009 //SYSPRINT DD  SYSOUT=*
```

Printing the Standard

To print a standard defined to CA JCLCheck, execute CAZ2JCL member Z1JCSRPT. The following JCL prints standard TESTSTD. Note that the standard name is specified in the PROC symbolic, STDNAME, on the EXEC statement.

```
//*********************************************************************
//* PRODUCE A PRINTOUT OF A SPECIFIC STANDARD FROM YOUR
//* STANDARDS DATABASE
//*********************************************************************
//PRTSTD        EXEC CAZ1RLST,
//              CAILOAD='CAI.CAILOAD',          <== LOADLIB WITH CAZ1RLST PGM
//              STDNAME='STDNAME',              <== STANDARD NAME TO BE PRINTED
//              RULES='CAI.JCLCHK.STANDARD'             <== JCS RULES DATABASE
```

This is how the standard looks when it is printed out.

```
                                        JOB CONTROL STANDARDS                                    PAGE      1
CA JCLCHECK             SYS: xxxx       STANDARD NAME: TESTSTD                  THURSDAY JANUARY 9, yyyy

*** TESTSTD  JOB/NAME

INITIAL: JOB NAME  NE T* ERR100

*** TESTSTD  ERROR MESSAGES

ERR100:  ERROR SEVERITY: E  ERROR MESSAGE #: 100
         ERROR TEXT: ALL TEST JOBS MUST HAVE A JOB NAME BEGI
         ERROR TEXT: NNING WITH T
```

Adding a Standard with More Than One Rule

In this example, you create a standard that is called PRODSTD in the Standards Rules Database. This standard directs CA JCLCheck to check for production jobs and requires a JOB NAME that begins with P; otherwise, it issues an error message. Next, regardless of the results of the first test, CA JCLCheck checks for the program name IMASPZAP and then the step name SMPAPPLY. When the program name is IMASPZAP, then the step name must equal SMPAPPLY; otherwise, it is an error. (When the program name is not IMASPZAP, CA JCLCheck does not check the step name because the program name test does not match.)

You use the following steps to define the criteria for the PRODSTD standards rules.

Here are the basic steps for adding a standard with more than one rule:

1. Create the standard name
2. Select the statement and parameter type
3. Define the standards rule
4. Select the action
5. Confirm the standard
6. Select the statement and parameter/subparameter type
7. Define the standards rule
8. Select the action
9. Define the standards rule
10. Define the action
11. Confirm the standard
12. Compile the standard
13. Use the standard
14. Print the standard

To access the JOB CONTROL STANDARD panels, enter **%JCKSTD** on the command line, and press Enter.

Creating the Standard Name

Use this panel to begin creating the new standard PRODSTD.

1. Enter **E** in the OPTION field.
2. Enter **PRODSTD** in the Standard Group Name field.

   Press Enter to move on to the next panel.

   For more information about masks, see How to Use CA JCLCheck (see page 13).

```
CAZ1RPSP ------------CA JCLCheck: STANDARDS (SELECTION)  ---------------
OPTION ===> E

Blank                 - Display Standard directory
E EDIT        - Edit/Create a Standard
D DELETE      - Delete a Standard
C COPY        - Copy a Standard
R RENAME      - Rename a Standard

Masks
  * - Match any String
  ? - Match any Character

Standard Group Name   ==> PRODSTD
```

Selecting the Statement and Parameter Type

The STANDARDS UPDATE panel (CAZ1RTSP) displays the allowable statement and parameter types for which you can define a JCL standard. Use this panel to define a rule for PRODSTD that relates to the JOB NAME statement and parameter type.

To define the first rule for the PRODSTD standard, enter **S** in the CMD field preceding JOB NAME, and press Enter to select the statement and parameter type JOB NAME.

```
CAZ1RTSP ------  CA JCLCheck: STANDARDS (UPDATE)  ------------- ROW 46 OF 61
COMMAND ===>                                                  SCROLL ===> PAGE


      STANDARD GROUP NAME : PRODSTD

   To select a Statement Type  :  Enter 'S' in CMD field, press Enter
   To return to previous Panel :  Press End

      CMD      TYPE     PARAMETER        SUB-PARAMETER
               EXEC     TIME_1           MINUTES
               EXEC     TIME_2           SECONDS
               JOB      ACCOUNT          NUMBER
               JOB      ADDRSPC
               JOB      CLASS
               JOB      GROUP
               JOB      MSGCLASS
       S       JOB      NAME
               JOB      NOTIFY
               JOB      PERFORM
               JOB      PGMR
               JOB      PRTY
               JOB      REGION
               JOB      TIME_1           MINUTES
               JOB      TIME_2           SECONDS
```

Defining the Standards Rule

You are now on the STANDARDS RULE DEFINITION panel (CAZ1R03P). Use this panel to define rules consisting of a relational operator, a value for that operator, and an action name. The action name, its operator, and value result in a specific action, which occurs during standards checking.

Use this panel to have the PRODSTD standard direct CA JCLCheck to produce error ERR200 if the job name is not equal to P.

1. Enter **NE** (not equal to) in the OPER field.
2. Enter **P** in the Value field.
3. Enter **ERR200** in the Action field. (You can name the action anything you like, in accordance with your company's standards.)

   Press End to move on to the next step.

```
CAZ1R03P ------ CA JCLCheck: STANDARDS (RULE DEFINITION) ---------- ROW 1 OF 1
COMMAND ===>                                                SCROLL ===> PAGE

  STANDARD GROUP NAME   : PRODSTD       CMD Field Usage:
  STATEMENT TYPE        : JOB            R - Replicate Line, Press Enter
  PARAMETER TYPE        : NAME           I - Insert Line, Press Enter
  SUB PARAMETER TYPE    :                D - Delete Line, Press Enter
  ACTION NAME           : INITIAL        S - Select Existing Action, press Enter
                Press End Returns to Previous Panel

       CMD     OPER    VALUE   ACTION
                 NE      P     ERR200
 ****************************** BOTTOM OF DATA **********************
```

Selecting the Action

The RELATED STANDARDS UPDATE panel (CAZ1RXSP) is the panel on which you select the statement and parameter type to define the action to be taken for this rule.

In this case, you want to select ERROR DEFINITION to define an error message.

- Enter **S** in the CMD field preceding ERROR DEFINITION, and press Enter.

```
 CAZ1RXSP -----  CA JCLCheck: RELATED STANDARDS (UPDATE)  ------  ROW 1 OF 63
 COMMAND ===>                                                SCROLL ===> PAGE

 To select a Statement Type  :  Enter 'S' in CMD field, press Enter
 To return to previous Panel :  Press End

      STANDARD GROUP NAME :    PRODSTD          ACTION : ERR100

      CMD      TYPE     PARAMETER        SUB-PARAMETER
               ERROR    DEFINITION       FAST PATH
               JOB      NAME
               =======  =============    =================
               DD       AVGREC
               DD       COPIES
               DD       DATACLAS
```

```
CAZ1RXSP ------  CA JCLCheck: RELATED STANDARDS (UPDATE) ------  ROW 31 OF 63
COMMAND ===>                                                  SCROLL ===> PAGE

To select a Statement Type :    Enter 'S' in CMD field, press Enter
To return to previous panel:    Press End

       STANDARD GROUP NAME :    PRODSTD          ACTION:ERR200

       CMD      TYPE     PARAMETER        SUB-PARAMETER
                ERROR    DEFINITION       FAST PATH
                DD       SYSOUT           CLASS
                DD       SYSOUT_2         WRITER
                DD       SYSOUT_3         FORM
                DD       UCS_1            CODE
                DD       UNIT
                DD       VOLUME_5         SERIAL_#
         S      ERROR    DEFINITION
                EXEC     ACCOUNT          INFO
                EXEC     ADDRSPC
```

Defining the Action

Use the STANDARDS DEFINITION panel (CAZ1RERP) to define the error message. The information entered here is the message you see whenever this error occurs.

1. Enter **E** in the Severity field.
2. Enter **200** in the Message# field.
3. Enter ALL PRODUCTION JOBS MUST HAVE A JOB NAME BEGINNING WITH P* in the Text field, and press End.

To define the error message for ERR200:

Note: When you press END, CA JCLCheck backs out through all previous Rule Definition panels. Continue to press End until the STANDARDS CONFIRMATION panel appears.

```
CAZ1RERP ---------  CA JCLCheck: STANDARDS (DEFINITION)  ------------------
COMMAND ===>                                                SCROLL ===> PAGE


      Delete Indicator:
      To delete Error enter D and press End

  STANDARD GROUP NAME   : PRODSTD
  STATEMENT TYPE        : ERROR
  PARAMETER TYPE        : DEFINITION
  ACTION NAME           : ERR200

  SEVERITY      : E   ( I - INFORMATIONAL, W - WARNING, E - ERROR, S - SEVERE)
  MESSAGE#      : 200
  TEXT          : ALL PRODUCTION JOBS MUST HAVE A JOB NAME BEGINNING WITH P*
```

Confirming the Standard

The STANDARDS CONFIRMATION panel (CAZ1RUCP) adds the standard to the Rules Database or cancels your request.

1. Press End to add TESTSTD to the Standard Rules database.
2. Press Cancel to cancel the request.

```
CAZ1RUCP ---------  CA JCLCheck: STANDARDS (CONFIRMATION)  ----------------
COMMAND ===>                                              SCROLL ===> PAGE


      Standard Name:   PRODSTD


   Press End to update the above standard on the rule database
   Use   CANCEL command to cancel the request
```

The STANDARDS (UPDATE) panel (CAZ1RTSP) is now redisplayed with JOB NAME above the divider line, indicating that this statement and parameter type is now included in the standard.

To define a rule to check the program name and step name, move on to the next step.

```
CAZ1RTSP --------  CA JCLCheck: STANDARDS (UPDATE)  -----------  ROW 1 OF 62
COMMAND ===>                                              SCROLL ===> PAGE

     To select a Statement Type :    Enter 'S' in CMD field, press Enter
     To return to previous panel:   Press End

       STANDARD GROUP NAME :    PRODSTD          ACTION:ERR200

       CMD      TYPE     PARAMETER        SUB-PARAMETER
                JOB      NAME
                =====    ==========     ================
                DD       AVGREC
                DD       COPIES
                DD       DATACLAS
```

Selecting a Statement and Parameter/Subparameter Type

To define another rule for PRODSTD to check the program name and step name, CA JCLCheck uses what is known as *procedure checking*. CA JCLCheck checks the new statement and parameter types regardless of the results of previous tests (JOB NAME). So program name is checked regardless of whether an error was encountered in JOB NAME, because it is considered a new rule path.

1. Scroll forward on panel CAZ1RTSP to view the statement and parameter type EXEC PGM.
2. Enter **S** preceding EXEC PGM in the CMD field, and press Enter.

```
CAZ1RTSP ----  CA JCLCheck: RELATED STANDARDS (UPDATE) -------  ROW 31 OF 62
COMMAND ===>                                                  SCROLL ===> PAGE


      To select a Statement Type :    Enter 'S' in CMD field, press Enter
      To return to previous panel:   Press End

      CMD      TYPE     PARAMETER        SUB-PARAMETER
               DD       SYSOUT           CLASS
               DD       SYSOUT_2         WRITER
               DD       SYSOUT_3         FORM
               DD       UCS_1            CODE
               DD       UNIT
               DD       VOLUME_5         SERIAL_#
               EXEC     ACCOUNT          INFO
               EXEC     ADDRSPC
               EXEC     DPRTY_1          VALUE1
               EXEC     DPRTY_2          VALUE2
               EXEC     DYNAMNBR
               EXEC     PERFORM
        S      EXEC     PGM
               EXEC     PROCNAME
               EXEC     REGION
```

Defining a Standards Rule

Use the STANDARDS RULE DEFINITION panel (CAZ1R06P) to define the rules (relational operator, value, and action name) for EXEC PGM and its resulting action.

For standard PRODSTD to direct CA JCLCheck to check for all program names that are equal to IMASZAP:

1. Enter **EQ (equal to)** in the OPER field.
2. Enter **IMASPZAP** in the Value field.
3. Enter **ACT@001** in the Action field. (You can name the action anything you like, in accordance with your company's standards.)

   Press End when you finish filling in this panel to move on to the next step.

```
CAZ1R06P ---- CA JCLCheck: STANDARDS (DEFINITION) --------------  ROW 1 OF 1
COMMAND ===>                                                  SCROLL ===> PAGE

  STANDARD GROUP NAME   : PRODSTD        CMD Field Usage:
  STATEMENT TYPE        : EXEC            R - Replicate Line, Press Enter
  PARAMETER TYPE        : PGM             I - Insert Line, Press Enter
  SUB PARAMETER TYPE    :                 D - Delete Line, Press Enter
  ACTION NAME           : INITIAL         S - Select Existing Action, Press Enter
               Press End Returns to Previous Panel

       CMD      OPER     VALUE      ACTION
                 EQ      IMASZAP    ACT@001
********************************** BOTTOM OF DATA **********************
```

Selecting the Action

Use the RELATED STANDARDS (UPDATE) panel (CAZ1RXSP) to select the statement and parameter type for EXEC STEPNAME.

When you access this panel after pressing End (in Step 8), panel CAZ1RXSP displays EXEC PGM above the divider line (========). The statement and parameter types begin at Row 1.

- Scroll down and enter **S** in the CMD field preceding EXEC STEPNAME, and press Enter.

```
CAZ1RXSP ------  CA JCLCheck: STANDARDS (UPDATE)  -------------  ROW 1 OF 63
COMMAND ===>                                                  SCROLL ===> PAGE

   To select a Statement Type :    Enter 'S' in CMD field, press Enter
   To return to previous panel:    Press End

       STANDARD GROUP NAME :     PRODSTD         ACTION : ACT@001

       CMD      TYPE     PARAMETER         SUB-PARAMETER
                EXEC     PGM
                =====    ============      =================
                DD       AVGREC
                DD       COPIES
```

```
CAZ1RXSP ---  CA JCLCheck: RELATED STANDARDS (UPDATE) --------  ROW 46 OF 63
COMMAND ===>                                                  SCROLL ===> PAGE

   To select a Statement Type :    Enter 'S' in CMD field, press Enter
   To return to previous panel:   Press End

       STANDARD GROUP NAME :    PRODSTD          ACTION : ACT@001

       CMD      TYPE     PARAMETER         SUB-PARAMETER
        S       EXEC     STEPNAME
                EXEC     TIME_1            MINUTES
                EXEC     TIME_2            SECONDS
                JOB      ACCOUNT           NUMBER
                JOB      ADDRSPC
```

Defining the Standards Rule

Use this panel to continue defining the action ACT@001 for PRODSTD.

To define the criteria for the error associated with EXEC PGM and EXEC STEPNAME:

1. Enter **NE (not equal to)** in the OPER field.
2. Enter **SMPAPPLY** in the Value field.

3. Enter **ERR201** in the Action field. (You can name the action anything you like, in accordance with your company's standards.)

   Press End when you finish filling in this panel to move on to the next step.

```
CAZ1R08P ---- CA JCLCheck: STANDARDS (RULE DEFINITION) ---------  ROW 1 OF 1
COMMAND ===>                                                 SCROLL ===> PAGE

   STANDARD GROUP NAME  :PRODSTD         CMD Field Usage:
   STATEMENT TYPE       : EXEC            R - Replicate Line, Press Enter
   PARAMETER TYPE       : STEPNAME        I - Insert Line, Press Enter
   SUB PARAMETER TYPE   :                 D - Delete Line, Press Enter
   ACTION NAME          : ACT@001         S - Select Existing Action, Press Enter
               Press End Returns to Previous Panel

       CMD      OPER     VALUE       ACTION
                 NE      SMPAPPLY    ERR201
```

Defining the Action

Use the STANDARDS DEFINITION PANEL (CAZ1RERP) to define the error message. The information entered here is the information that is displayed when this error occurs.

To define error message ERR201:

1. Enter **E** in the Severity field.
2. Enter **201** in the Message# field.
3. Enter FOR PRODUCTION JOBS PROGRAM IMASPZAP CAN ONLY EXECUTE IN STEP SMPAPPLY in the Text field, and press End.

   Note: When you press End, CA JCLCheck backs out through all previous Rule Definition panels. Continue to press End until the UPDATE CONFIRMATION panel appears.

```
CAZ1RERP ---------  CA JCLCheck: STANDARDS (DEFINITION)  ------------------
COMMAND ===>                                                 SCROLL ===> PAGE


    Delete Indicator:
    To delete Error enter D and press End

  STANDARD GROUP NAME   : PRODSTD
  STATEMENT TYPE        : ERROR
  PARAMETER TYPE        : DEFINITION
  ACTION NAME           : ERR201

  SEVERITY : E   ( I - INFORMATIONAL, W - WARNING, E - ERROR, S - SEVERE)
  MESSAGE# : 201
  TEXT  : FOR PRODUCTION JOBS PROGRAM IMASPZAP CAN ONLY EXECUTE IN STEP SMPAPPLY
```

Confirming the Standard

The Standards Confirmation panel (CAZ1RUCP) adds standard PRODSTD to the Standard Rules database or cancels your request.

1. Press End to add PRODSTD to the Standard Rules database.
2. Press Cancel to cancel the request.

Although the PRODSTD standard was added to the database when JOB NAME was defined, it is added again (or updated) to the database with the new rule path.

```
CAZ1RUCP -------  CA JCLCheck: STANDARDS (CONFIRMATION)  ----------------
 COMMAND ===>                                           SCROLL ===> PAGE


      Standard Name:     PRODSTD


   Press End to update the above standard on the rule database
   Use   CANCEL command to cancel the request
```

Compiling the Standard

To compile the standard PRODSTD, repeat the compile procedure in this section "Compiling the Standard". Adding a Standard contains the EXEC PROC CAZ1JCS step, the EXEC PROC CAZ1ASM step and the EXEC PROC CAZ1LNK step.

Keep in mind that when you compile the standard, the evaluation program CAZ1EVAL compiles *every* standard in the database. If you are adding several standards at once, it is best to wait until they are all added before compiling.

Using the Standard

To use this standard, specify the runtime option STANDARD (PRODSTD) when you execute CA JCLCheck.

Printing the Standard

This is what the standard looks like when it is printed using CAZ1RLST:

```
                                  JOB CONTROL STANDARDS                                                PAGE       1
CA JCLCHECK             SYS: xxxx      STANDARD NAME: PRODSTD                                 THURSDAY JANUARY 9, 20yy

*** PRODSTD  EXEC/PGM

INITIAL: EXEC PGM  EQ IMASZAP ACT@001
    ACT@001: EXEC STEPNAME  NE SMPAPPLY ERR201

*** PRODSTD  JOB/NAME

INITIAL: JOB NAME  NE  P*  ERR200
*** PRODSTD  ERROR MESSAGES
ERR200:  ERROR SEVERITY: E  ERROR MESSAGE #: 000
                  ERROR TEXT: 200
                  ERROR TEXT: ALL PRODUCTION JOBS MUST HAVE A JOB NAME
                  ERROR TEXT: BEGINNING WITH P
ERR201:  ERROR SEVERITY: E  ERROR MESSAGE #: 201
                  ERROR TEXT: FOR PRODUCTION JOBS PROGRAM IMASPZAP CA
                  ERROR TEXT: N ONLY EXECUTE IN STEP SMPAPPLY
```

Deleting a Standard

The STANDARDS DELETE CONFIRMATION panel (CAZ1RDCP) deletes the standard from the Standards Rules database.

1. On the Standards Selection panel (CAZ1RPSP), enter **D** in the OPTION field.
2. Do one of the following:
   - Enter the Standard Group Name in the Standard Group Name field Group Name field, and press Enter.
   - On the Standard Directory panel (CAZ1RDSP), enter **D** in the CMD field preceding the Standard Group Name, and press Enter.
3. This displays the Standards Delete Confirmation panel (CAZ1RDCP) with the name of the selected standard. Press Enter to delete the standard, or End to exit without deleting the definition.

```
CAZ1RPSP -------------CA JCLCheck: STANDARDS (SELECTION)  --------------
 OPTION ===> D

Blank                - Display Standard directory
E EDIT       - Edit/Create a Standard
D DELETE     - Delete a Standard
C COPY       - Copy a Standard
R RENAME     - Rename a Standard

 Masks
   * - Match any String
   ? - Match any Character

 Standard Group Name  ==> TESTSTD
```

```
CAZ1RDSP ----------- CA JCLCheck: STANDARDS (DIRECTORY)   ------------------
COMMAND ===>                                                  SCROLL ===>

  OPT: E    - Edit a Standard
       D    - Delete a Standard
       C    - Copy a Standard
       R    - Rename a Standard

              OPT      STANDARD-GROUP          UPDATED          USER ID
               D       TESTSTD                 yyyy/mm/dd 15:12 USER02
                       PRODSTD                 yyyy/mm/dd 09:22 USER02
```

```
CAZ1RDCP --------  CA JCLCheck: STANDARDS (CONFIRMATION)  ---------------
COMMAND ===>                                          SCROLL ===> PAGE


    Standard Name:      TESTSTD


    Press Enter to delete the above standard from the rule database
          End to cancel the request
```

Deleting a Rule Related to an Error Message

To delete a standards rule related to an error message, perform the following steps to delete the message and the rule.

1. Access the STANDARD SELECTION panel (CAZ1RPSP), Creating the Standard Name (see page 127), and enter **E** in the OPTION field.
2. Do one of the following:
   - Enter a standard name in the Standard Group Name field, and press Enter.
   - Leave all fields blank on the STANDARDS (SELECTION) panel and press Enter. At the Standard Directory panel, enter **E** in the CMD field preceding the standard name, and press Enter.

```
 CAZ1RPSP ----------CA JCLCheck: STANDARDS (SELECTION)  -------------------
&col.OPTION ===> E

  Blank    - Display Standard directory
  E EDIT   - Edit/Create a Standard
  D DELETE - Delete a Standard
  C COPY   - Copy a Standard
  R RENAME - Rename a Standard

   Masks
     * - Match any String
     ? - Match any Character

   Standard Group Name       ==> PRODSTD
```

```
CAZ1RDSP ------ CA JCLCheck: STANDARDS (DIRECTORY)   ----------------------
COMMAND ===>                                                        SCROLL
===>

  OPT: E   - Edit a Standard
       D    - Delete a Standard
       C    - Copy a Standard
       R    - Rename a Standard

           OPT      STANDARD-GROUP    UPDATED                  USER ID
           TESTSTD                yyyy/mm/dd 15:12        USER02
            E       PRODSTD           yyyy/mm/dd 09:22         USER02
  ***************************** BOTTOM OF DATA ****************************
```

3. Enter **S** in the CMD field preceding the statement type and parameter on the STANDARDS UPDATE panel (CAZ1RTSP), Selecting the Statement and Parameter Type (see page 127), that applies to the rule you want to delete, and press Enter.

```
CAZ1RTSP --------  CA JCLCheck: STANDARDS (UPDATE)  ------------  ROW 1 OF 63
COMMAND ===>                                                 SCROLL ===> PAGE


           STANDARD GROUP NAME :      TESTSTD

   To select a Statement Type  :     Enter 'S' in CMD field, press Enter
   To return to previous panel :     Press End

   CMD      TYPE     PARAMETER         SUB-PARAMETER
             S       JOB      NAME
            EXEC     PGM
                     =====    ========      ==============
                     DD       AVGREC
                     DD       COPIES
                     DD       DATACLAS
                     DD       DCB_1             BLKSIZE
                     DD       DCB_2             LRECL
                     DD       DCB_3             RECFM
                     DD       DCB_4             KEYLEN
                     DD       DDNAME
                     DD       DEST_1            NODE
                     DD       DEST_2            USER
                     DD       DISP_1            STATUS
                     DD       DISP_2            NORMAL
```

4. The STANDARDS RULE DEFINITION panel for that rule is displayed, Defining the Standards Rule (see page 128). Enter **D** in the CMD field preceding the rule you want to delete.

```
CAZ1R03P ------ CA JCLCheck: STANDARDS (RULE DEFINITION) -------- ROW 1 OF 1
COMMAND ===>                                                  SCROLL ===> PAGE

  STANDARD GROUP NAME:  PRODSTD       CMD Field Usage:
  STATEMENT TYPE    :  JOB            R - Replicate Line, Press Enter
  PARAMETER TYPE    :  NAME           I - Insert Line, Press Enter
  SUB PARAMETER TYPE:                 D - Delete Line, Press Enter
  ACTION NAME       :  INITIAL        S - Select Existing Action, press Enter
           Press End Returns to Previous Panel

   CMD      OPER     VALUE    ACTION
    D         NE         P    ERR200
 ******************************** BOTTOM OF DATA ***************************
```

5. When the STANDARDS (CONFIRMATION) panel appears, press End to delete the rule from the standard.

6. If other rules exist for this parameter in this standard, you are finished deleting this rule. However, if there are no other rules for this parameter, CA JCLCheck displays another Confirmation panel. Press Enter to delete the rule from the standard. Press End again. The Standards Definition panel is displayed again with the message, STANDARD UPDATED.

```
CAZ1RUCP ---------  CA JCLCheck: STANDARDS (CONFIRMATION)  -----------------
COMMAND ===>                                                  SCROLL ===> PAGE


    Standard Name: PRODSTD


    Press End to update the above standard on the rule database
    Use   CANCEL command to cancel the request
```

```
CAZ1RECP ---------  CA JCLCheck: STANDARDS (CONFIRMATION)  ---------------
COMMAND ===>                                                  SCROLL ===> PAGE

      Standard Name:       PRODSTD

 During the EDIT all rules were deleted for the initial path:

         JOB  NAME


   Press Enter to delete this portion of the standard from the database
     End to cancel the request
```

Deleting Action Rules from a Standard

To delete a standards rule, you can delete the action for the rule, and the rule; or, if the action for the rule is used by more than one parameter, you can leave the action and simply delete the rule.

1. Access the STANDARD (SELECTION) panel (CAZ1RPSP), Creating the Standard Name (see page 127), and enter **E** in the OPTION field.
2. Do one of the following:
   - Enter a standard name in the Standard Group Name field, and press Enter.
   - At the Standard Directory panel, enter **E** in the CMD field preceding the standard name, and press Enter.

```
CAZ1RPSP ---------------CA JCLCheck: STANDARDS (SELECTION)  ---------------
   OPTION ===> E

  blank    - Display Standard directory
  E EDIT   - Edit/Create a Standard
  D DELETE - Delete a Standard
  C COPY   - Copy a Standard
  R RENAME - Rename a Standard

   Masks
     * - Match any String
     ? - Match any Character

   Standard Group Name       ==> PRODSTD
```

```
CAZ1RDSP ------------ CA JCLCheck: STANDARDS (DIRECTORY)   -----------------
COMMAND ===>                                                        SCROLL
===>

  OPT:     E    - Edit a Standard
           D    - Delete a Standard
           C    - Copy a Standard
           R    - Rename a Standard

   OPT             STANDARD-GROUP    UPDATED                 USER ID
                   TESTSTD           yyyy/mm/dd 15:12        USER02
    E              PRODSTD           yyyy/mm/dd 09:22        USER02
  ********************************** BOTTOM OF DATA **********************
```

3. Enter **S** in the CMD field preceding the statement type and parameter on the STANDARDS UPDATE panel (CAZ1RTSP) that applies to the rule you want to delete, and press Enter.

```
CAZ1RTSP -----------  CA JCLCheck: STANDARDS (UPDATE)  --------  ROW 1 OF 63
COMMAND ===>                                                    SCROLL ===> PAGE


 STANDARD GROUP NAME :      PRODSTD

  To select a Statement Type :    Enter 'S' in CMD field, press Enter
  To return to previous panel:    Press End

 CMD      TYPE     PARAMETER        SUB-PARAMETER
  S       EXEC     PGM
          EXEC     STEPNAME
                   ======  ==========     ================
                   DD      AVGREC
                   DD      COPIES
                   DD      DATACLAS
                   DD      DCB_1          BLKSIZE
                   DD      DCB_2          LRECL
                   DD      DCB_3          RECFM
                   DD      DCB_4          KEYLEN
                   DD      DDNAME
                   DD      DEST_1         NODE
                   DD      DEST_2         USER
                   DD      DISP_1         STATUS
                   DD      DISP_2         NORMAL
```

4. The STANDARDS RULE DEFINITION panel for that rule is displayed, Defining the Standards Rule (see page 128). Enter **S** in the CMD field preceding the rule you want to delete.

```
CAZ1R06P------  CA JCLCheck: STANDARDS (RULE DEFINITION)  ------  ROW 1 OF 1
COMMAND ===>                                                    SCROLL ===> DATA


 STANDARD GROUP NAME: PRODSTD       CMD Field Usage:
 STATEMENT TYPE     : EXEC           R - Replicate Line, Press Enter
 PARAMETER TYPE     : PGM            I - Insert Line, Press Enter
 SUB PARAMETER TYPE:                 D - Delete Line, Press Enter
 ACTION NAME        : INITIAL        S - Select Existing Action, Press Enter
                      Press End Returns to Previous Panel

 CMD      OPER     VALUE            ACTION
  S         EQ     IMASZAP          ACT@001
 ****************************** BOTTOM OF DATA ******************************
```

5. This displays the STANDARDS RULE DEFINITION panel again. This time it contains the statement and parameter type for the error message attached to the action for the rule that you want to delete. Enter **D** in the CMD field preceding this rule, and press End.

```
CAZ1R06P------  CA JCLCheck: STANDARDS (RULE DEFINITION)  -----  ROW 1 OF 1
COMMAND ===>                                                  SCROLL ===> DATA


 STANDARD GROUP NAME: PRODSTD        CMD Field Usage:
 STATEMENT TYPE     : EXEC            R - Replicate Line, Press Enter
 PARAMETER TYPE     : STEPNAME        I - Insert Line, Press Enter
 SUB PARAMETER TYPE:                  D - Delete Line, Press Enter
 ACTION NAME.       : ACT@001         S - Select Existing Action, Press Enter
                     Press End Returns to Previous Panel

 CMD      OPER     VALUE             ACTION
  D        NE      SMPAPPLY          ERR201
 ********************************* BOTTOM OF DATA ************************
```

6. The STANDARDS (RULE DEFINITION) panel appears again, containing the rule you want to delete, with question marks (?) in the ACTION rule field. Enter **D** preceding the action, and press End. This deletes the rule.

```
CAZ1R06P------  CA JCLCheck: STANDARDS (RULE DEFINITION)  -----  ROW 1 OF 1
COMMAND ===>                                                  SCROLL ===> DATA


 STANDARD GROUP NAME: PRODSTD        CMD Field Usage:
 STATEMENT TYPE     : EXEC            R - Replicate Line, Press Enter
 PARAMETER TYPE     : PGM             I - Insert Line, Press Enter
 SUB PARAMETER TYPE:                  D - Delete Line, Press Enter
 ACTION NAME        : INITIAL         S - Select Existing Action, Press Enter
                     Press End Returns to Previous Panel

 CMD      OPER     VALUE    ACTION
  D        EQ      IMASZAP  ????????
 ********************************* BOTTOM OF DATA ************************
```

7. The STANDARDS CONFIRMATION panel is displayed. Press End to delete the standard.

8. Press End again. If this statement and parameter type have no other rules, another Confirmation panel appears. Press Enter to delete this statement and parameter type as a rule from this standard on the Standard Rules database

   Note: If more than one rule exists using this statement and parameter type, you bypass this Confirmation panel.

   When the confirmation is complete you are returned to the Standards Definition panel, which displays the message, "STANDARD UPDATED."

```
CAZ1RUCP --------  CA JCLCheck: STANDARDS (CONFIRMATION)  -----------------
COMMAND ===>                                                SCROLL ===> PAGE


          Standard Name:  PRODSTD


     Press End to update the above standard on the rule database
     Use   CANCEL command to cancel the request
```

```
----------------  CA JCLCheck: STANDARDS (CONFIRMATION)  ---------------
COMMAND ===>                                                SCROLL ===> DATA


          Standard Name: PRODSTD

During the EDIT all rules were deleted for the initial path:
          EXEC  PGM


Press Enter to delete this portion of the standard from the database
  End to cancel the request
```

Chapter 4: CA JCLCheck REXX Programming Interface

The REXX programming language allows the Standards Administrator to define JCL site standards for compliance/enforcement checking by CA JCLCheck.

The Standards Administrator should have a fundamental understanding of the REXX programming language to use this interface. If you are the administrator and are unfamiliar with REXX, you may be able to use the coding convention from CAZ1REXX, the sample REXX EXECs supplied in the CAZ2CLS0 library.

Important! For information about the REXX Function Calls used in the REXX programming language, see functions in the z/OS TSO/E REXX Reference Manual.

This section contains the following topics:

REXX EXECs (see page 145)
REXX EXEC #1 (see page 147)
REXX EXEC #2 (see page 156)

REXX EXECs

The administrator *must* tailor a copy of the CA JCLCheck sample REXX EXEC template (CAZ1REXX) to create any REXX EXEC because the sample templates contain variables that provide an interface to CA JCLCheck.

Note: For information about the REXX interface and a basic tutorial of the REXX EXEC, see REXX for CA JCLCheck and JCLNeat in the *Programming Guide*.

Accessing the CAZ1REXX Template

Make a copy of CAI.CAZ2CLS0 (CAZ1REXX) to tailor REXX EXEC #1 and REXX EXEC #2.

Creating a REXX EXEC

Read the steps on the following pages and enter the numbered lines of code into your copy of the CAZ1REXX EXEC.

Note: Since you do not have to modify either the INITIAL PROCESSING and DO WHILE loop or the $CA_JCLERROR subroutine, they are not included as part of the steps in the program.

Submitting JCL Using the REXX EXEC

Copy your edited version of the CAZ1REXX EXEC into a library referenced by the SYSEXEC DD statements tailored during product installation. When the member is in the proper library, specify the STDREXX option, STDREXX(*rexx exec member name*) for CA JCLCheck, and submit a JCL member to CA JCLCheck processing to test the REXX EXEC.

There are example output reports (Report #2 and Report #6) at the end of each example. These reports display REXX error messages that were generated by the REXX EXEC during standards processing of a JCL member.

The INITIAL PROCESSING and DO WHILE Loop

Important! Do not remove or modify any lines prior to the Initial Processing Subroutine. All variables prefixed with $CA are required.

The following initialization and main DO WHILE loop demonstrate the external function calls available for the REXX EXEC. The CAZ1JRXI interface passes the proper values to the REXX EXEC variables. The interface directs the processing flow throughout execution for each statement type encountered in the JCL member you submit for standards processing.

```
/*********************************REXX*********************************/
/*  CA JCLCheck       REXX programming interface for JCLNeat/       */
/**************************************************************************/
/*  Initialization logic                                                  */
/**************************************************************************/
PARSE ARG $CAJCL_PARM;              /* Get argument */
/*                                                                        */
Call INITIAL_PROCESSING                                                   */
/*                                                                        */
/*      Continually call the appropriate statement subroutine            */
/*      until there are no more jobs to process                          */
/*                                                                        */
DO WHILE $CAJCL_REASON ¬= 'EOR'
$CAJCL_VARS:
  If $CAJCL_REASON = 'JOB' then CALL JOB_PROCESSING
   If $CAJCL_REASON = 'EXE' then CALL EXEC_PROCESSING
    If $CAJCL_REASON = 'DD' then CALL DD_PROCESSING
     If $CAJCL_REASON = 'EOS' then CALL END_STEP_PROCESSING
      If $CAJCL_REASON = 'EOJ' then CALL END_JOB_PROCESSING
       If $CAJCL_REASON = 'RAW' then CALL RAW_DATA_PROCESSING
        If $CAJCL_REASON = 'MSG' then CALL MSG_PROCESSING;
  CALL 'CAZ1JRXI' $CAJCL_PARM       /* Call the interface*/
 $CAJCL_REASON = RESULT;           /* Save result in $CAJCL_REASON  */
End
EXIT
```

The CALL INITIAL_PROCESSING statement executes a subroutine to set any user-defined variables to be used in this EXEC.

The DO WHILE loop executes the appropriate subroutine function as the JCL is processed. The INITIAL PROCESSING statement and DO WHILE loop contain $CAJCL statements. These are required and should not be changed.

Note: For more information, see REXX for CA JCLCheck and JCLNeat in the *Programming Guide*.

The $CAJCL_ERROR Subroutine

This subroutine is called from subsequent subroutines to issue the associated error messages. These messages carry the prefix CAY6341.

```
/*********************************************************************/
/*  Internal subroutine to issue a CA JCLCheck error message         */
/*********************************************************************/
/*  Prior to calling set the values of:                              */
/*  $CAJCL_SEVERITY must contain the severity of message: I,W,E,S    */
/*     I=informational, W=warning, E=error, S=serious                */
/*  $CAJCL_MESSAGE must contain the text of the message, any case    */
/*  can be specified. (upper/lower). The resultant message is        */
/*  prefixed with "CAY6341".                                         */
/*********************************************************************/
$CAJCL_ERROR:
PARSE ARG $CAJCL_SEVERITY,$CAJCL_MESSAGE
  CALL 'CAZ1JRXE' $CAJCL_PARM $CAJCL_SEVERITY $CAJCL_MESSAGE
Return
```

You do not need to modify this subroutine. It calls all the necessary CA JCLCheck internal functions needed to produce user-generated error messages for JCL.

Note: If you supply an invalid severity code, CA JCLCheck assumes a severity I (informational).

REXX EXEC #1

In this EXEC you modify a copy of the REXX EXEC template (CAZ1REXX) to create a simple standard that accomplishes the following tasks:

- Ensure that jobs submitted to class O run only after 5 p.m.
- Monitor the number of cartridge drives
- Count the number of (E)rror messages issued

- Conditionally issue a (S)evere error message for number of (E)rror level messages
- Count the number of (S)evere error messages issued
- Track the utilization of DD statements

If the JCL you submit for standards checking does not meet the criteria above, CA JCLCheck issues an error message defined within the criteria.

Modifying the INITIAL PROCESSING Subroutine

CA JCLCheck executes the Initial Processing Subroutine once per execution. This code allows the Standards Administrator to set initial values for any user-defined variables to be used in the EXEC.

1. Set the initial value of a counter for cart (cartridges) to zero for subsequent subroutines that use the *cart_count* variable.
2. Set the initial value of a counter for dd (DD statements) to zero for subsequent subroutines that use the *dd_count* variable.
3. Set the initial value of a counter for error ((E)rror messages) to zero for subsequent subroutines that use the *error_count* variable.

```
/*********************************************************************/
/*  Initial Processing Subroutine                                    */
/*********************************************************************/
INITIAL_PROCESSING:
 cart_count = 0
 dd_count = 0
 error_count = 0
Return
```

Modifying the JOB Processing Subroutine

Note: All values for the statement variables (that is, JOB, EXEC, DD) change when they encounter a new value for that statement variable.

Change the JOB Statement subroutine to obtain the current time and date. Code an IF statement test so that if the time is less than 17:00 hours (5:00 p.m.) and the job class is equal to O, then CA JCLCheck issues an error message if the test is true.

1. Assign the current hour to the *curr_hour* variable.
2. Assign the current time in civil format, *HH:MMnn*, with *nn* being a.m. or p.m., to the *curr_c_time* variable.
3. Code the IF statement on the following line. If the JOB.CLASS variable is equal to O and the current time is less than 17:00 hours (5:00 p.m.), the statements within the following DO loop are executed.

4. Code the DO loop to call the $CAJCL_ERROR subroutine so that CA JCLCheck issues the associated (E)rror message.

   The following are error message severity levels:

   I = 0

   W = 4

   E = 8

   S = 12

   This adheres to normal CA JCLCheck error severity standards.

```
/*********************************************************************/
/*  Job Card Processing Subroutine                                   */
/*********************************************************************/
JOB_PROCESSING:
curr_hour = TIME(H)     /*   Get the current hour */
curr_c_time = TIME(C)     /*   Get time in Civil format */
If JOB.CLASS = 'O' & curr_hour < 17 then
  Do
 Call $CAJCL_ERROR,
 'E','Class "O" is reserved for second shift usage, retry after 5 PM, it is
 'only curr_c_time
 End
Return
```

Modifying the EXEC Processing Subroutine

Modify this subroutine so that any trailing blanks in the step name on the EXEC statement are removed. Removing the blanks gives the error message a neater appearance.

To modify the EXEC processing subroutine, assign the non-blank value of EXEC.STEPNAME to the *step_name* variable.

```
/*********************************************************************/
/*  Exec Card processing Subroutine                                  */
/*********************************************************************/
EXEC_PROCESSING:
 step_name = STRIP(EXEC.STEPNAME,T)       /* Remove trailing blanks */
Return
```

Modifying the DD Processing Subroutine

Modify this DD processing subroutine to increment the DD statement counter. If the DD statement UNIT parameter is CART (cartridge), have the cart count statement increment the count by 1 to keep track of the number of cartridges used per step.

1. Increment the *dd_count* variable by one each time a DD statement is processed.
2. Code the IF statement to check for a value equal to CART (cartridge) in the DD statement UNIT parameter. If the parameter is equal to CART, the following statements within the DO loop are executed.
3. Code the DO loop to increment the cartridge count by 1.

```
/*********************************************************************/
/*  Data Definition Processing Subroutine                            */
/*********************************************************************/
DD_PROCESSING:
 dd_count = (dd_count + 1)
If (DD.UNIT = 'CART') then
 Do
 cart_count = (cart_count + 1)
 End
Return
```

Modifying the END-OF-STEP Processing Subroutine

Change this subroutine so that CA JCLCheck:

- Issues a warning message if the number of cartridges in this JCL step exceeds four.
- Issues an error message if the number of DD statements in this JCL step exceeds ten.
- Keeps track of the number of error messages invoked during the execution of this JCL step.

1. Code the first IF statement to determine the number of cartridges found within the current step. If the number is greater than four, the following statements within the DO loop are executed.
2. Code the DO loop on the following line to call the $CAJCL_ERROR subroutine so that CA JCLCheck issues the associated (W)arning message.
3. Code the second IF statement to determine the number of DD statements found in the current step. If the number is greater than ten, the following statements within the DO loop are executed.
4. Code the DO loop on the following line to call the $CAJCL_ERROR subroutine so that CA JCLCheck issues the associated (E)rror message.

5. Code the *error_count*= (*error_count* +1) to increment the number of errors by 1 upon issuance of an (E) severity code or higher.

   Note: The levels of severity code are assigned as follows: I = 0, W = 4, E = 8, S = 12. This adheres to normal CA JCLCheck error severity.

6. Code *dd_count* variable=0 to reset the DD counter to zero for use if there is another step in the JCL.

7. Code *cart_count* variable=0 to reset the cartridge counter to 0 for use if there is another step.

```
/*********************************************************************/
/*  Step End  processing Subroutine                                  */
/*********************************************************************/
END_STEP_PROCESSING:
If (cart_count > 4) then
 Do
 Call $CAJCL_ERROR,
 'W','Step 'step_name' is using 'cart_count' cartridge drives, please
reduce utilization'
 End
If (dd_count > 10) then
 Do
 Call $CAJCL_ERROR,
 'E','Step 'step_name' has specified 'dd_count' DD names, this exceeds site
standards'
 error_count = (error_count + 1)
 End
dd_count = 0
cart_count = 0
Return
```

Modifying the END-OF-JOB Processing Subroutine

Modify this subroutine to test the number of errors that occurred during processing of this JCL stream. If there were errors, have CA JCLCheck issue a severe error message to prevent job submission.

1. Code this IF statement to examine the number of errors found in the current job. If the number is greater than 0, the following statements within the DO loop are executed.

2. Code this DO loop to call the $CAJCL_ERROR subroutine so that CA JCLCheck issues the associated (S)evere error message.

```
/*********************************************************************/
/*  Job  End  processing Subroutine                                  */
/*********************************************************************/
END_JOB_PROCESSING:
IF error_count > 0 Then
 Do
   Call $CAJCL_ERROR,
 'S','JOB 'JOB.JOBNAME' has encountered too many site violations. Execution
has been aborted'
 End
Return
```

Examining Report 2 - Listing of Merged JCL

Examine this report to see a listing of the entire JCL for the job, including procedure statements. This report displays the JCL that was processed using the STDREXX option with the REXX EXEC in this sample.

```
CA                                  REPORT 2 - LISTING OF MERGED JCL
CA JCLCHECK             SYS: xxxx   JOB: PMK00100          PGMR: CAUSER                FRIDAY SEPTEMBER 10, yyyy
-------------------------------------------------------------------------------------------------------------------
  STMT   STEP      PROCSTEP   ERR       CMNT
  NUM    NAME      NAME       SEV       COL       STATEMENT TEXT
-------------------------------------------------------------------------------------------------------------------

   1.                          8        //PMK00100 JOB (40100000),'S. BELL',CLASS=O,MSGCLASS=X,
                                        //    MSGLEVEL=(1,1),PERFORM=255,REGION=5120K,
                                        //    PRTY=15,TIME=(357900,10),
                                        //    NOTIFY=IBMUSER,ADDRSPC=VIRT,USER=USER02
   2.    STEP1                          //STEP1    EXEC PGM=IEFBR14
   3.    STEP1                          //PSDD1    DD  DSN=USER02.JCL.CNTL,DISP=OLD
   4.    STEP2                          //STEP2 EXEC PGM=AMASPZAP,PERFORM=111
   5.    STEP2                          //SYSPRINT DD SYSOUT=*
   6.    STEP2                          //SYSUT3   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
   7.    STEP2                          //SYSUT4   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
   8.    STEP2                4         //SYSIN    DD DUMMY
   9.    STEP3                          //STEP3    EXEC PGM=IEFBR14,ADDRSPC=REAL
  10.    STEP3                4         //DD1 DD UNIT=CART,DISP=(,PASS)
  11.    STEP3                4         //DD2 DD UNIT=CART,DISP=(,PASS)
  12.    STEP3                4         //DD3 DD UNIT=CART,DISP=(,PASS)
  13.    STEP3                4         //DD4 DD UNIT=CART,DISP=(,PASS)
  14.    STEP3                4         //DD5 DD UNIT=CART,DISP=(,PASS)
  15.    STEP3                4         //DD6 DD UNIT=CART,DISP=(,PASS)
  16.    STEP4                8         //STEP4    EXEC PGM=IEFBR14
  17.    STEP4                4         //DD1 DD UNIT=TAPE,DISP=(,PASS)
  18.    STEP4                4         //DD2 DD UNIT=TAPE,DISP=(,PASS)
  19.    STEP4                4         //DD3 DD UNIT=TAPE,DISP=(,PASS)
  20.    STEP4                4         //DD4 DD UNIT=TAPE,DISP=(,PASS)
  21.    STEP4                4         //DD5 DD UNIT=TAPE,DISP=(,PASS)
  22.    STEP4                4         //DD6 DD UNIT=TAPE,DISP=(,PASS)
  23.    STEP4                8         //DD7 DD UNIT=SYSDA,DISP=(,PASS)
  24.    STEP4                8         //DD8 DD UNIT=SYSDA,DISP=(,PASS)
  25.    STEP4                8         //DD9 DD UNIT=SYSDA,DISP=(,PASS)
  26.    STEP4                8         //DDA DD UNIT=SYSDA,DISP=(,PASS)
  27.    STEP4                8         //DDB DD UNIT=SYSDA,DISP=(,PASS)
  28.    STEP5                          //STEP5    EXEC PGM=IEFBR14
  29.    STEP5                4         //DD1 DD UNIT=TAPE,DISP=(,PASS)
  30.    STEP5                4         //DD2 DD UNIT=TAPE,DISP=(,PASS)
  31.    STEP5                4         //DD3 DD UNIT=TAPE,DISP=(,PASS)
  32.    STEP5                4         //DD4 DD UNIT=TAPE,DISP=(,PASS)
  33.    STEP5               12         //
```

For more information on displaying CA JCLCheck reports using the STDREXX option with this EXEC, see Submitting JCL Using the REXX EXEC (see page 146).

Examining Report 6 - Error Messages

Examine this report to see a list of all error messages associated with the JCL member that was submitted for CA JCLCheck processing. CA JCLCheck only produces this report if it detects one or more errors. You can direct this report to appear at the beginning or end of the report output or incorporate it into Report 2 - Listing of Merged JCL, depending on the setting of the ERROR option.

```
                                  REPORT 6 - ERROR MESSAGES
CA JCLCHECK              SYS: xxxx       JOB: PMK00100       PGMR: CAUSER                                FRIDAY SEPTEMBER 10, yyyy
-------------------------------------------------------------------------------------------------------------------------------------
  STMT   MESSAGE
  NUM    NUMBER    MESSAGE TEXT
-------------------------------------------------------------------------------------------------------------------------------------

   1.    CAY6341E   'Class "O" is reserved for second shift usage, retry after 5 PM, it is only 4:50pm'
   8.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
   9.    CAY6341W   'STEP3         is using 6 cartridge drives, please reduce utilization'
  10.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
  11.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
  12.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
  13.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
  14.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
  15.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
  16.    CAY6341E   'Step STEP4 has specified 11 DD names, this exceeds site standards'
  17.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
  18.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
  19.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
  20.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
  21.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
  22.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
  23.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
               CAY6083E SPACE PARAMETER NOT PROVIDED FOR NEW DIRECT ACCESS DATASET
  24.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
               CAY6083E SPACE PARAMETER NOT PROVIDED FOR NEW DIRECT ACCESS DATASET
  25.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
               CAY6083E SPACE PARAMETER NOT PROVIDED FOR NEW DIRECT ACCESS DATASET
  26.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
               CAY6083E SPACE PARAMETER NOT PROVIDED FOR NEW DIRECT ACCESS DATASET
  27.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
               CAY6083E SPACE PARAMETER NOT PROVIDED FOR NEW DIRECT ACCESS DATASET
  29.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
  30.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
  31.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
  32.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
  33.    CAY6341S   'JOB PMK00100 has encountered too many site violations. Execution has been aborted'
CAY6000 26 STATEMENTS FLAGGED IN JOB 'PMK00100' MAXIMUM SEVERITY WAS 12
```

The following messages were flagged by REXX EXEC #1 during standards processing.

STMT NUM 1.

CAY6341E 'Class "O" is reserved for second shift usage, retry after 5 PM, it is only 4:50 pm.'

STMT NUM 9.

CAY6341W 'Step STEP3 is using 6 cartridge drives, please reduce utilization.'

STMT NUM 16.

CAY6341E 'Step STEP4 has specified 11 ddnames, this exceeds site standards.'

STMT NUM 33.

CAY6341S 'JOB PMK00100 has encountered too many site violations. Execution has been aborted.'

Sample REXX Output Using EDCHEK

This is a partial sample of REXX output for this JCL using EDCHEK.

```
 EDIT ------- USER02.JCL.CNTL(SB43) - 01.00 -------------------------- LAST CC WAS  12
 COMMAND ===>                                                         SCROLL ===> CSR
 ****** ******************************** TOP OF DATA ********************************
==MSG>  CAY6000 25 STATEMENTS FLAGGED IN JOB "PMK00100" MAXIMUM SEVERITY WAS 12
==MSG>
==MSG>
==MSG>
  000100 //PMK00100 JOB (40100000),'S. BELL',CLASS=O,MSGCLASS=X,
==MSG>   //PMK00100 JOB (40100000),"S. BELL",CLASS=O,MSGCLASS=X,
  000110 //    MSGLEVEL=(1,1),PERFORM=255,REGION=5120K,
==MSG>   //    MSGLEVEL=(1,1),PERFORM=255,REGION=5120K,
  000111 //    PRTY=15,TIME=(357900,10),
==MSG>   //    PRTY=15,TIME=(357900,10),
  000112 //    NOTIFY=IBMUSER,ADDRSPC=VIRT,USER=USER02
==MSG>   //    NOTIFY=IBMUSER,ADDRSPC=VIRT,USER=USER02
==MSG>  CAY6341E "CLASS "O" IS RESERVED FOR SECOND SHIFT USAGE, RETRY AFTER 7
==MSG>        PM, IT IS ONLY 5:21PM"
==MSG>  CAY6341S "JOB PMK00100 HAS ENCOUNTERED TOO MANY SITE VIOLATIONS
==MSG>        EXECUTION HAS BEEN ABORTED"
==MSG>
  000113 //STEP1    EXEC PGM=IEFBR14
  000114 //PSDD1    DD  DSN=USER02.JCL.CNTL,DISP=OLD
  000115 //STEP2 EXEC PGM=AMASPZAP,PERFORM=111
  000116 //SYSPRINT DD SYSOUT=*
  000117 //SYSUT3   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
  000118 //SYSUT4   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
  000119 //SYSIN    DD DUMMY
==MSG>   //SYSIN    DD DUMMY
==MSG>  CAY6087W BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL
==MSG>         S013 IN EXECUTION
==MSG>
  000120 //STEP3    EXEC PGM=IEFBR14,ADDRSPC=REAL
==MSG>   //STEP3    EXEC PGM=IEFBR14,ADDRSPC=REAL
 ==MSG> CAY6341W "STEP STEP3 IS USING 6 CARTRIDGE DRIVES, PLEASE REDUCE
==MSG>         UTILIZATION"
==MSG>
  000130 //DD1 DD UNIT=CART,DISP=(,PASS)
==MSG>   //DD1 DD UNIT=CART,DISP=(,PASS)
==MSG>  CAY6087W BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL
==MSG>         S013 IN EXECUTION
==MSG>
  000140 //DD2 DD UNIT=CART,DISP=(,PASS)
==MSG>   //DD2 DD UNIT=CART,DISP=(,PASS)
==MSG>  CAY6087W BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL
==MSG>         S013 IN EXECUTION
==MSG>
  000150 //DD3 DD UNIT=CART,DISP=(,PASS)
==MSG>   //DD3 DD UNIT=CART,DISP=(,PASS)
==MSG>  CAY6087W BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL
==MSG>         S013 IN EXECUTION
==MSG>
  000160 //DD4 DD UNIT=CART,DISP=(,PASS)
==MSG>   //DD4 DD UNIT=CART,DISP=(,PASS)
==MSG>  CAY6087W BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL
==MSG>         S013 IN EXECUTION
==MSG>
  000170 //DD5 DD UNIT=CART,DISP=(,PASS)
==MSG>   //DD5 DD UNIT=CART,DISP=(,PASS)
==MSG>  CAY6087W BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL
==MSG>         S013 IN EXECUTION
******************************** BOTTOM OF DATA ********************************
```

REXX EXEC #2

If you created REXX EXEC #1, use it as the template to create this EXEC; otherwise, use a copy of the CAZ1REXX EXEC supplied in the CA JCLCheck CAZ2CLS0 library.

Note: For information about template access, formatting the EXEC, submitting JCL, and the main DO WHILE and $CAJCL_Error Subroutine sections of the EXEC, see REXX EXEC #1.

In this EXEC you modify the REXX EXEC to create a standard that is more complex than REXX EXEC #1. The new standard accomplishes the following:

- Reports on production jobs (with PMK job name prefix) not submitted to class P
- Reports on technical services jobs (with SYS job name prefix) not submitted to class X
- Permits address space equal to REAL only on the job statement for authorized user (user ID is SYS)
- Prevents use of the PERFORM parameter on the EXEC statement
- Permits address space equal to REAL on the EXEC statement for authorized user
- Ensures that program prefix PS is being used with user ID CASCHED

If the JCL you submit for standards checking does not meet the previous criteria, CA JCLCheck issues an error message defined within the criteria.

Modifying the INITIAL PROCESSING Subroutine

Modify the Initial Processing Subroutine so that CA JCLCheck sets the count on the number of cartridges, DD statements, and error messages to zero before processing any of the JCL.

1. Set the initial value of a counter for cart (cartridges) to zero for subsequent subroutines that use the *cart_count* variable.
2. Set the initial value of a counter for dd (DD statements) to zero for subsequent subroutines that use the *dd_count* variable.
3. Set the initial value of a counter for error ((E)rror messages) to zero for subsequent subroutines that use the *error_count* variable.

```
/*********************************************************************/
/*  Initial Processing Subroutine                                    */
/*********************************************************************/
INITIAL_PROCESSING:
 cart_count = 0
 dd_count = 0
 error_count = 0
Return
```

Modifying the JOB Processing Subroutine

Note: All values for the statement variables (for example, JOB, EXEC, DD) change when they encounter a new value for that statement variable.

Change this subroutine so that:

- The job name prefix is three characters long.
- The job's user ID prefix is three characters long.
- If the job prefix is PMK and its class is not P, then CA JCLCheck issues a warning message.
- If the job prefix is SYS and its class is not X, then CA JCLCheck issues a warning message.
- If the address space is REAL and the prefix of the user ID on the job statement is not SYS, CA JCLCheck issues an error message that names this parameter as a restricted parameter, increment the error count by one, and prevent job submission.

1. Code the first statement to extract the first three characters of the job name and store them in this *job_prefix* variable.

   Note: If this job name is fewer than three characters, the job prefix variable is filled with trailing spaces.

2. Code the statement on the following line to extract the first three characters of the user ID and store them in this *user_prefix* variable.
3. Code the IF statement on the following line so that if the job_prefix value is equal to PMK and the job class is not equal to P, the statements within the following DO loop are executed.
4. Code the following DO loop to call the $CAJCL_ERROR subroutine so that CA JCLCheck issues the associated (W)arning message.
5. Code the IF statement on the following line so that if the job_prefix value is equal to SYS and the job class is not equal to X, the following statements within the DO loop are executed.
6. Code the following DO loop to call the $CAJCL_ERROR subroutine so that CA JCLCheck issues the associated (W)arning message.

7. Code the IF statement on the following line so that if the address space value is equal to real and the user prefix is not equal to SYS on the JOB card, the following statements within the DO loop are executed.
8. Code the following DO loop to call the $CAJCL_ERROR subroutine so that CA JCLCheck issues the associated (E)rror message.
9. Code the *error_count*= (*error_count* + 1) to increment the number of errors by 1 upon the issuance of an (E) severity code level or higher.

   Note: The levels of severity codes are assigned as follows:

   I = 0, W = 4, E = 8, S = 12.

   This adheres to normal CA JCLCheck error severity.

```
/*********************************************************************/
/*  Job Card Processing Subroutine                                   */
/*********************************************************************/
JOB_PROCESSING:
job_prefix = SUBSTR(JOB.JOBNAME,1,3)
user_prefix = SUBSTR(JOB.USERID,1,3)
If job_prefix = 'PMK' & JOB.CLASS ¢ 'P' then
 Do
 Call $CAJCL_ERROR,
 'W','Class "P" should be used for production jobs'
 End
If job_prefix = 'SYS' & JOB.CLASS ¢ 'X' then
 Do
 Call $CAJCL_ERROR,
 'W','Class "X" should be used for technical services jobs'
 End
If JOB.ADDRSPC = 'REAL' & user_prefix ¢ 'SYS' then
 Do
 Call $CAJCL_ERROR,
 'E','ADDRSPC=REAL is a restricted JCL parameter'
 error_count = (error_count + 1)
 End
Return
```

Modifying EXEC Processing Subroutine

Modify this subroutine so that:

- The job's program name prefix on the EXEC statement is two characters long.
- The job's step name on the EXEC statement does not contain any trailing blank spaces in an error message.
- If the EXEC statement contains a PERFORM= operand, the PERFORM parameter is greater than 0. CA JCLCheck issues an error message prohibiting use of the PERFORM parameter, and increment the error message count by one.

- If the EXEC statement specifies address space is REAL and the user ID prefix is not SYS, CA JCLCheck issues an error message that the REAL address space is a restricted parameter, and increment the error message count by one.
- If the job's user ID is not CASCHED and the program prefix is PS, CA JCLCheck issues an error message that only production programs can be run by the automated scheduler, and prevent job submission.

1. Set the *pgm_prefix* variable equal to the first two characters of the program name.
2. Code the *step_name* variable to strip out any trailing blank spaces in the EXEC.STEPNAME variable for use in an EOS error message.
3. Code the IF statement on the following line to check for a value greater than 0. This value represents the EXEC statement PERFORM parameter. If the value is greater than 0, the following statements within the DO loop are executed.
4. Code the following DO loop to call the $CAJCL_ERROR subroutine so that CA JCLCheck issues the associated (E)rror message.
5. Code the *error_count*= (*error_count* + 1) to increment the number of errors by 1 upon the issuance of an (E) severity code level or higher.
6. Code the IF statement on the following line so that if the address space value is equal to REAL on the EXEC statement and the user ID prefix on the JOB card is not equal to SYS, the following statements within the DO loop are executed.
7. Code the following DO loop to call the $CAJCL_ERROR subroutine so that CA JCLCheck issues the associated (E)rror message.
8. Code *error_count*= (*error_count* + 1) to increment the number of errors by 1 upon the issuance of an (E) severity code level or higher.
9. Code this IF statement so that if the pgm_prefix value is equal to PS and the job user ID is not equal to CASCHED, the following statements within the DO loop are executed.

10. Enter the following DO loop to call the $CAJCL_ERROR subroutine so that CA JCLCheck issues the associated (S)evere message.

    Note: Error_count is not incremented by 1 for this message because it is not a severity (E) or higher.

```
/*********************************************************************/
/*  Exec Card processing Subroutine                                  */
/*********************************************************************/
  EXEC_PROCESSING:
pgm_prefix = SUBSTR(EXEC.PGM,1,2)
step_name = STRIP(EXEC.STEPNAME,T)       /* Remove trailing blanks */
If EXEC.PERFORM > 0 then
 Do
 Call $CAJCL_ERROR,
 'E','PERFORM parameter is not allowed, remove and rerun'
 error_count = (error_count + 1)
 End
If EXEC.ADDRSPC = 'REAL' & user_prefix ¢ 'SYS' then
 Do
 Call $CAJCL_ERROR,
 'E','ADDRSPC=REAL is a restricted JCL parameter'
 error_count = (error_count + 1)
 End
 If (JOB.USERID ¢ 'CASCHED') & (pgm_prefix = 'PS') then
  Do
  Call $CAJCL_ERROR,
 'S','PRODUCTION PROGRAMS CAN ONLY BE RUN BY THE AUTOMATED SCHEDULER'
 End
Return
```

Modifying the DD Processing Subroutine

Change this subroutine to increment the DD statement counter by one each time a DD statement is encountered and count the number of DD statement cartridge UNIT parameters and increment the count by one.

1. Code this statement to count the number of DD statements in the submitted JCL. This variable is incremented by one each time a DD statement is processed.
2. Code this IF statement to check for a value equal to CART (cartridge) in the DD statement UNIT parameter. If the parameter is equal to CART, the following statements within the DO loop are executed.
3. Code the DO loop to increment the cartridge count by 1.

```
/*********************************************************************/
/*  Data Definition Processing Subroutine                            */
/*********************************************************************/
DD_PROCESSING:
 dd_count = (dd_count + 1)
If (DD.UNIT = 'CART') then
 Do
 cart_count = (cart_count + 1)
 End
Return
```

Modifying the END-OF-STEP Processing Subroutine

Change the EOS processing subroutine so that:

- If the number of cartridges is greater than four, CA JCLCheck issues a warning message per step to reduce cartridge drive utilization.
- If the number of DD statements exceeds 10, CA JCLCheck issues an error message that the STEP has exceeded site standards, increment the number of error messages by one and prevent job submission.

1. Code the first IF statement to determine the number of cartridges found within the current step. If the number is greater than four, the following statements within the DO loop are executed.
2. Code the following DO loop to call the $CAJCL_ERROR subroutine so that CA JCLCheck issues the associated (W)arning message.
3. Code the IF statement on the following line to determine the number of DD statements found in the current step. If the number is greater than ten, the following statements within the DO loop are executed.
4. Code the following DO loop to call the $CAJCL_ERROR subroutine so that CA JCLCheck issues the associated (E)rror message.
5. Code *error_count*= (*error_count* + 1) to increment the number of errors by 1 upon the issuance of an (E) severity code level or higher.

   Note: The levels of severity codes are assigned as follows:

   I = 0, W = 4, E = 8, S = 12.

   This adheres to normal CA JCLCheck error severity.
6. Reset the DD counter and CART counter.
7. Code *cart_count* variable=0 to reset the cartridge counter to 0 for next step processing.

```
/*********************************************************************/
/*  Step End  processing Subroutine                                  */
/*********************************************************************/
END_STEP_PROCESSING:
If (cart_count > 4) then
 Do
 Call $CAJCL_ERROR,
 'W','Step 'step_name' is using 'cart_count' cartridge drives, please
reduce utilization'
 End
If (dd_count > 10) then
 Do
 Call $CAJCL_ERROR,
 'E','Step 'step_name' has specified 'dd_count' DD names, this exceeds site
standards'
 error_count = (error_count + 1)
 End
 dd_count = 0
 cart_count = 0
Return
```

Modifying the END-OF-JOB Processing Subroutine

Modify this subroutine so that if there are any error messages CA JCLCheck issues a severe error against the job's job name preventing job submission.

1. Code the IF statement to determine the number of errors found in the current job. If the number is greater than 0, the following statements within the DO loop are executed.
2. Code the DO loop to call the $CAJCL_ERROR subroutine so that CA JCLCheck issues the associated (S)evere error message.

```
 /*********************************************************************/
 /*  Job  End  processing Subroutine                                  */
 /*********************************************************************/
 END_JOB_PROCESSING:
If error_count > 0 then
 Do
Call $CAJCL_ERROR,
'S','JOB 'JOB.JOBNAME' has encountered too many site violations execution
has been aborted';
 End
Return
```

Examining Report 2 - Listing of Merged JCL

Examine this report to see a listing of the entire JCL for the job, including procedure statements. This report displays the JCL that was processed, in a step-by-step format using this STDREXX EXEC.

```
                                    REPORT 2 - LISTING OF MERGED JCL
CA JCLCHECK              SYS: xxxx      JOB: PMK00100      PGMR: CAUSER           FRIDAY SEPTEMBER 10, yyyy
-------------------------------------------------------------------------------------------------------------------------
  STMT   STEP       PROCSTEP   ERR        CMNT
  NUM    NAME       NAME       SEV        COL       STATEMENT TEXT
-------------------------------------------------------------------------------------------------------------------------

   1.                           4         //PMK00100 JOB (40100000),'S. BELL',CLASS=X,MSGCLASS=X,
                                          //              MSGLEVEL=(1,1),PERFORM=255,REGION=5120K,
                                          //              PRTY=15,TIME=(357900,10),
                                          //              NOTIFY=IBMUSER,ADDRSPC=VIRT,USER=USER02
   2.    STEP1                           //STEP1       EXEC PGM=IEFBR14
   3.    STEP1                           //PSDD1      DD  DSN=USER02.JCL.CNTL,DISP=OLD
   4.    STEP2                8          //STEP2 EXEC PGM=AMASPZAP,PERFORM=111
   5.    STEP2                           //SYSPRINT DD SYSOUT=*
   6.    STEP2                           //SYSUT3   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
   7.    STEP2                           //SYSUT4   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
   8.    STEP2                4          //SYSIN      DD DUMMY
   9.    STEP3                8          //STEP3      EXEC PGM=IEFBR14,ADDRSPC=REAL
  10.    STEP3                4          //DD1 DD UNIT=CART,DISP=(,PASS)
  11.    STEP3                4          //DD2 DD UNIT=CART,DISP=(,PASS)
  12.    STEP3                4          //DD3 DD UNIT=CART,DISP=(,PASS)
  13.    STEP3                4          //DD4 DD UNIT=CART,DISP=(,PASS)
  14.    STEP3                4          //DD5 DD UNIT=CART,DISP=(,PASS)
  15.    STEP3                4          //DD6 DD UNIT=CART,DISP=(,PASS)
  16.    STEP4                4          //STEP4      EXEC PGM=IEFBR14
  17.    STEP4                4          //DD1 DD UNIT=TAPE,DISP=(,PASS)
  18.    STEP4                4          //DD2 DD UNIT=TAPE,DISP=(,PASS)
  19.    STEP4                4          //DD3 DD UNIT=TAPE,DISP=(,PASS)
  20.    STEP4                4          //DD4 DD UNIT=TAPE,DISP=(,PASS)
  21.    STEP4                4          //DD5 DD UNIT=TAPE,DISP=(,PASS)
  22.    STEP4                4          //DD6 DD UNIT=TAPE,DISP=(,PASS)
  23.    STEP4                8          //DD7 DD UNIT=SYSDA,DISP=(,PASS)
  24.    STEP4                8          //DD8 DD UNIT=SYSDA,DISP=(,PASS)
  25.    STEP4                8          //DD9 DD UNIT=SYSDA,DISP=(,PASS)
  26.    STEP4                8          //DDA DD UNIT=SYSDA,DISP=(,PASS)
  27.    STEP4                8          //DDB DD UNIT=SYSDA,DISP=(,PASS)
  28.    STEP5                8          //STEP5      EXEC PGM=IEFBR14
  29.    STEP5                4          //DD1 DD UNIT=TAPE,DISP=(,PASS)
  30.    STEP5                4          //DD2 DD UNIT=TAPE,DISP=(,PASS)
  31.    STEP5                4          //DD3 DD UNIT=TAPE,DISP=(,PASS)
  32.    STEP5                4          //DD4 DD UNIT=TAPE,DISP=(,PASS)
  33.    STEP6                8          //STEP6      EXEC PGM=PSTEST9
  33.    STEP6                12         //
```

Note: For information about displaying this report, see REXX for CA JCLCheck and JCLNeat in the *Programming Guide*.

Examining Report 6 - Error Messages

Examine this report to see a list of all error messages associated with this job. CA JCLCheck only produces this report if it detects one or more errors. You can direct this report to appear at the beginning or end of the report output or incorporate it with REPORT 2 - LISTING OF MERGED JCL, depending on the setting of the ERROR option.

```
                                  REPORT 6 - ERROR MESSAGES
CA JCLCHECK             SYS: xxxx      JOB: PMK00100     PGMR: CAUSER                      FRIDAY SEPTEMBER 10, yyyy
-------------------------------------------------------------------------------------------------------------------------------
  STMT   MESSAGE
  NUM    NUMBER    MESSAGE TEXT
-------------------------------------------------------------------------------------------------------------------------------

   1.    CAY6341W   'Class "P" should be used for production jobs'
   4.    CAY6341E   'PERFORM parameter is not allowed, remove and rerun'
   8.    CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
   9.    CAY6341E   'ADDRSPC=REAL is a restricted JCL parameter'
               CAY6341W 'Step STEP3 is using 6 cartridge drives, please reduce utilization'
   10.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
   11.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
   12.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
   13.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
   14.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
   15.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
   16.   CAY6341E   'Step STEP4 has specified 11 DD names, this exceeds site standards'
   17.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
   18.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
   19.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
   20.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
   21.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
   22.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
   23.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
               CAY6083E SPACE PARAMETER NOT PROVIDED FOR NEW DIRECT ACCESS DATASET
   24.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
               CAY6083E SPACE PARAMETER NOT PROVIDED FOR NEW DIRECT ACCESS DATASET
   25.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
               CAY6083E SPACE PARAMETER NOT PROVIDED FOR NEW DIRECT ACCESS DATASET
   26.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
               CAY6083E SPACE PARAMETER NOT PROVIDED FOR NEW DIRECT ACCESS DATASET
   27.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
               CAY6083E SPACE PARAMETER NOT PROVIDED FOR NEW DIRECT ACCESS DATASET
   29.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
   30.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
   31.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
   32.   CAY6087W   BLKSIZE NOT SPECIFIED FOR NEW OR DUMMY DATA SET - POTENTIAL S013 IN EXECUTION
   33.   CAY6341I   'PRODUCTION PROGRAMS CAN ONLY BE RUN BY THE AUTOMATED SCHEDULER'
               CAY6093E PROGRAM 'PSTEST9' NOT FOUND
   34.   CAY6341S 'JOB PMK00100 has encountered too many site violations execution has been aborted'
CAY6000 27 STATEMENTS FLAGGED IN JOB 'PMK00100 MAXIMUM SEVERITY WAS 12
```

The following are the messages that were flagged by the REXX EXEC during standards processing.

STMT NUM 1.

CAY6341W 'Class "P" should be used for production jobs'

STMT NUM 4.

CAY6341E 'PERFORM parameter is not allowed, remove and rerun'

STMT NUM 9.

CAY6341E 'ADDRSPC=REAL is a restricted JCL parameter'

CAY6341W 'Step STEP3 is using 6 cartridge drives, please reduce utilization'

STMT NUM 16.

CAY6341E 'Step STEP4 has specified 11 DD names, this exceeds site standards'

STMT NUM 33.

CAY6341I 'PRODUCTION PROGRAMS CAN ONLY BE RUN BY THE AUTOMATED SCHEDULER'

STMT NUM 34.

CAY6341S 'JOB PMK00100 has encountered too many site violations execution has been aborted'

Chapter 5: JCLNeat REXX Programming Interface

The REXX programming language allows the Standards Administrator to define JCL site standards for reformatting by JCLNeat.

The Standards Administrator should have a fundamental understanding of the REXX programming language to use this interface. If you are the administrator and are unfamiliar with REXX, you may be able to use the coding convention from CAZ1NREX, the sample REXX EXEC supplied in the CAZ2CLS0 library.

Important! For information about the REXX Function Calls used in the REXX programming language, see functions in the TSO Extensions Version 2 Procedures Language MVS/REXX Reference Manual.

This section contains the following topics:

REXX EXECs (see page 167)
The INITIAL PROCESSING and DO WHILE Loop (see page 168)
REXX EXEC (see page 170)
Reformatted JCL Report (see page 178)

REXX EXECs

The administrator *must* tailor a copy of the JCLNeat sample REXX EXEC template (CAZ1NREX) to create any REXX EXEC because the sample template contains variables that provide an interface to JCLNeat.

Note: For information about the REXX interface and a basic tutorial of the REXX EXEC, see REXX for CA JCLCheck and JCLNeat in the *Programming Guide*.

Accessing the CAZ1NREX Template

Make a copy of CAI.CAZ2CLS0 (CAZ1NREX) to tailor for this exercise.

Creating a REXX EXEC

Read the steps on the following pages and enter the numbered lines of code into your copy of the CAZ1NREX EXEC.

Note: Since you do not have to modify either the INITIAL PROCESSING and DO WHILE loop or the $CA_JCLERROR subroutine, they are not included as part of the steps in the program.

Submitting JCL Using the REXX EXEC

Copy your edited version of the CAZ1NREX EXEC into a library referenced by the SYSEXEC DD statements tailored during product installation. When the member is in the proper library, specify the REXXMEM option, REXXMEM=*rexx exec member name* and submit a JCL member to JCLNeat processing to test the REXX EXEC.

There are example output reports (Report #2 and Report #6) at the end of each example. These reports display REXX error messages that were generated by the REXX EXEC during standards processing of a JCL member.

The INITIAL PROCESSING and DO WHILE Loop

Important! Do not remove or modify any lines prior to the Initial Processing Subroutine. All variables prefixed with $CA are required.

The initialization and main DO WHILE loop following, demonstrate the external function calls available for the REXX EXEC. The CAZ1NRXI interface passes the proper values to the REXX EXEC variables. The interface directs the processing flow throughout execution for each statement type encountered in the JCL member you submit for standards processing.

```
/*********************************REXX*********************************/
/*  CA JCLCheck       REXX programming interface for JCLNeat        */
/**********************************************************************/
/*  Initialization logic                                              */
/**********************************************************************/
PARSE ARG $CAJCL_PARM;       /* Get argument */
/*                                                                    */
CALL INITIAL_PROCESSING      /* Do one time logic */
/*                                                                    */
/*      Continually call the appropriate statement subroutine       */
/*      until there are no more jobs to process                     */
/*                                                                    */
DO WHILE $CAJCL_REASON ¬= 'EOR'
$CAJCL_VARS:
  If $CAJCL_REASON = 'JOB' then CALL JOB_PROCESSING;
   If $CAJCL_REASON = 'EXE' then CALL EXEC_PROCESSING;
    If $CAJCL_REASON = 'DD' then CALL DD_PROCESSING;
     If $CAJCL_REASON = 'EOS' then CALL END_STEP_PROCESSING;
      If $CAJCL_REASON = 'EOJ' then CALL END_JOB_PROCESSING;
       If $CAJCL_REASON = 'RAW' then CALL RAW_DATA_PROCESSING;
  CALL 'CAZ1NRXI' $CAJCL_PARM ;
 $CAJCL_REASON = RESULT;           /* Save result in $CAJCL_REASON  */
End
EXIT
```

The CALL INITIAL_PROCESSING statement executes a subroutine to set any user-defined variables to be used in this EXEC.

The DO WHILE loop executes the appropriate subroutine function as the JCL is processed. The INITIAL PROCESSING and DO WHILE loop subroutines contain $CAJCL statements; they are required and should not be changed.

Note: For more information, see REXX for CA JCLCheck and JCLNeat in the *Programming Guide*.

The $CAJCL_ERROR Subroutine

This subroutine is called from subsequent subroutines to issue the associated error messages. These messages carry the prefix CAY6541.

```
/*********************************************************************/
/*  Internal subroutine to issue a CA JCLCheck error message         */
/*********************************************************************/
/*  Prior to calling set the values of:                              */
/*  $CAJCL_SEVERITY must contain the severity of message: I,W,E,S    */
/*     I=informational, W=warning, E=error, S=serious                */
/*  $CAJCL_MESSAGE must contain the text of the message, any case    */
/*  can be specified. (upper/lower). The resultant message is        */
/*  prefixed with "CAY6541".                                         */
/*********************************************************************/
$CAJCL_ERROR:
PARSE ARG $CAJCL_SEVERITY,$CAJCL_MESSAGE;
  CALL 'CAZ1NRXE' $CAJCL_PARM $CAJCL_SEVERITY $CAJCL_MESSAGE ;
Return
```

You do not need to modify this subroutine. It calls all the necessary JCLNeat internal functions needed to produce user-generated error messages for JCL.

Note: If you supply an invalid severity code, JCLNeat assumes a severity I (informational).

REXX EXEC

In this EXEC, you modify a copy of the REXX EXEC template (CAZ1NREX) to create a simple standard that accomplishes the following tasks:

- Ensure that jobs submitted to class O run only after 5 p.m. and if the job is submitted prior to 5:00 p.m., then change the JOB.CLASS to D
- Monitor the number of cartridge drives
- Count the number of (E)rror messages issued
- Conditionally issue a (S)evere error message for number of (E)rror level messages
- Count the number of (S)evere error messages issued
- Track the utilization of DD statements
- Examine unit names and ddnames that are no longer valid and tailor the JCL accordingly

If the JCL you submit for standards checking does not meet the previous criteria, CA JCLCheck issues an error message defined within the criteria.

Modifying the INITIAL PROCESSING Subroutine

CA JCLCheck executes the Initial Processing Subroutine once per execution. This code allows the Standards Administrator to set initial values for any user-defined variables to be used in the EXEC.

1. Set the initial value of a counter for cart (cartridges) to zero for subsequent subroutines that use the *cart_count* variable.
2. Set the initial value of a counter for dd (DD statements) to zero for subsequent subroutines that use the *dd_count* variable.
3. Set the initial value of a counter for error ((E)rror messages) to zero for subsequent subroutines that use the *error_count* variable.

```
/*********************************************************************/
/*  Initial Processing Subroutine                                    */
/*********************************************************************/
INITIAL_PROCESSING:
 cart_count = 0
 dd_count   = 0
 error_count = 0
Return
```

Modifying the JOB Processing Subroutine

Note: All values for the statement variables (that is, JOB, EXEC, DD) change when they encounter a new value for that statement variable.

Change the JOB Statement subroutine to obtain the current time and date. Code an IF statement test so that if the time is less than 17:00 hours (5:00 pm) and the job class is equal to O, CA JCLCheck issues an error message if the test is true.

1. Assign the current hour to the *curr_hour* variable.
2. Assign the current time in civil format, *HH:MMnn*, with *nn* being a.m. or p.m., to the *curr_c_time* variable.
3. Code the IF statement. If the JOB.CLASS variable is equal to O and the current time is less than 17:00 hours (5:00 p.m.), the statements within the following DO loop are executed.
4. Code the DO loop to call the $CAJCL_ERROR subroutine so that JCLNeat issues the associated (E)rror message, then set job class to D.

   Note: Error message severity levels:

   I = 0, W = 4, E = 8, S = 12

   This adheres to normal CA JCLCheck error severity standards.

```
/*********************************************************************/
/*  Job Card Processing Subroutine                                   */
/*********************************************************************/
JOB_PROCESSING:
 curr_hour = TIME(H)      /* Get current hour */
 curr_c_time = TIME(C)      /* Get time in civil format */
 If JOB.CLASS = 'O' & curr_hour < 17 then
 Do
 Call $CAJCL_ERROR,
 'W','Class "O" is reserved for second shift usage, retry after 5PM,
 it is only 'curr_c_time 'class set for first shift use'
JOB.CLASS = 'D'
End
Return
```

Modifying the EXEC Processing Subroutine

Modify this subroutine so that any trailing blanks in the step name on the EXEC statement are removed. Removing the blanks gives the error message a neater appearance.

To modify the EXEC processing subroutine, assign the non-blank value of EXEC.STEPNAME to the step_name variable.

```
 /*********************************************************************/
 /* Exec Card processing Subroutine                                   */
/*********************************************************************/
EXEC_PROCESSING:
 step_name = STRIP(EXEC.STEPNAME,T)   /* Remove trailing blanks *
 Return
```

Modifying the DD Processing Subroutine

Modify this DD processing subroutine to increment the DD statement counter. If the DD statement UNIT parameter is CART (cartridge), have the cart count statement increment the count by 1 to keep track of the number of cartridges used per step.

1. Increment the *dd_count* variable by one by 1 each time a DD statement is processed.
2. Code the IF statement on the following line to check for a value equal to CART (cartridge) in the DD statement UNIT parameter. If the parameter is equal to CART, the following statements within the DO loop are executed.

3. Code the following DO loop to increment the cartridge count by 1.
4. Code the IF statement on the following line to check the value of UNIT for the name BOGUS and if the unit name is BOGUS, execute the statements in the following DO WHILE loop.
5. Code the following DO loop to issue an informational message and set the value of UNIT to null, removing it from the JCL.
6. Code the IF statement on the following line to check the value of ddname for the name OBSOLETE and if the ddname is OBSOLETE, execute the statements in the following DO loop.
7. Code the following DO loop to issue a warning message and set the delete_flag to Y for later processing in the raw_processing subroutine.
8. Code the IF statement on the following line to check the value of ddname for the name COMMENT and if the ddname is COMMENT, execute the statements in the following DO loop.
9. Code the following DO loop to issue an informational message and set the comment_flag to Y for later processing in the raw_processing subroutine.

```
/*********************************************************************/
/*  Data Definition Processing Subroutine                            */
/*********************************************************************/
DD_PROCESSING:
 dd_count = (dd_count + 1)
 If (DD.UNIT = 'CART') then
  Do
 cart_count = (cart_count + 1)
 End
 If (DD.UNIT = 'BOGUS') then
  Do
 Call $CAJCL_ERROR,
 'I','Unit name coded 'dd.unit' is no longer valid and will be removed'
 DD.UNIT = ''                        /* remove unit name from JCL */
 End
 If (DD.DDNAME = 'OBSOLETE') then
  Do
 delete_flag = 'Y'                    /* delete this statement */
 Call $CAJCL_ERROR,
 'W','DD named 'dd.ddname' was coded and is no longer required, this statement
 will be deleted'
 End
 If (DD.DDNAME = 'COMMENT') then
  Do
 comment_flag = 'Y'                   /* delete this statement */
 Call $CAJCL_ERROR,
 'I','DD named 'dd.ddname' will be commented out, it is no longer valid'
End
Return
```

Modifying the END-OF-STEP Processing Subroutine

Change this subroutine so that CA JCLCheck:

- Issues an error message if the number of cartridges in this JCL step exceeds four.
- Issues an error message if the number of DD statements in this JCL step exceeds ten.
- Keeps track of the number of error messages invoked during the execution of this JCL step.

1. Code the first IF statement to determine the number of cartridges found within the current step. If the number is greater than four, the following statements within the DO loop are executed.
2. Code the following DO loop to call the $CAJCL_ERROR subroutine so that CA JCLCheck issues the associated (W)arning message.
3. Code the *error_count*= (*error_count* + 1) to increment the number of errors by 1 upon the issuance of an (E) severity code level or higher.

   Note: The levels of severity code are assigned as follows: I = 0, W = 4, E = 8, S = 12. This adheres to normal CA JCLCheck error severity.
4. Code the IF statement on the following line to determine the number of DD statements found in the current step. If the number is greater than 10, the following statements within the DO loop are executed.
5. Code the following DO loop to call the $CAJCL_ERROR subroutine so that CA JCLCheck issues the associated (E)rror message.
6. Code the *error_count*= (*error_count* + 1) to increment the number of errors by 1 upon the issuance of an (E) severity code level or higher.

   Note: The levels of severity code are assigned as follows: I = 0, W = 4, E = 8, S = 12. This adheres to normal CA JCLCheck error severity.
7. Code *dd_count* variable=0 to reset the DD counter to zero for use if there is another step in the JCL.
8. Code error *cart_count* variable=0 to reset the cartridge counter to 0 for use if there is another step.

```
  /*********************************************************************/
  /*  Step End  processing Logic                                       */
  /*********************************************************************/
  END_STEP_PROCESSING:
   If (cart_count > 4) then
   Do
   Call $CAJCL_ERROR,
   'E','Step 'step_name' is using 'cart_count' cartridge drives, please
reduce 'Utilization'
     error_count = (error_count + 1)
  End
  If (dd_count > 10) then
  Do
   Call $CAJCL_ERROR,
   'E','Step 'step_name' has specified 'dd_count' DD names, this exceeds
site 'Standards'
    error_count = (error_count + 1)
  End
   dd_count = 0
   cart_count = 0
 Return
```

Modifying the END-OF-JOB Processing Subroutine

Modify this subroutine to test the number of errors that occurred during processing of this JCL stream. If there were errors, have CA JCLCheck issue a severe error message to prevent job submission.

1. Code the IF statement to examine the number of errors found in the current job. If the number is greater than 0, the following statements within the DO loop are executed.
2. Code the DO loop to call the $CAJCL_ERROR subroutine so that CA JCLCheck issues the associated (S)evere error message.
3. Code error *cart_count* variable=0 to reset the cartridge counter to 0 for use if there is another step.

```
  /*********************************************************************/
  /*  Job  End  processing Logic                                       */
  /*********************************************************************/
  END_JOB_PROCESSING:
  If error_count > 0 Then
   Do
     Call $CAJCL_ERROR,
     'S','JOB 'JOB.JOBNAME' has encountered too many site violations'
  error_count = 0
  End
    Return
```

RAW Data Processing Subroutine

Modify this subroutine to check for statements that need to be deleted from or commented out of the JCL.

1. Code the first IF statement to determine whether the delete_flag was set during dd_processing. If the delete flag is set to Y, execute the following statements in the DO loop.
2. Code the following DO loop to set the record count to 0 to indicate that this statement is to be deleted. The DO loop sets the delete flag back to N after the statement is deleted.
3. Code the IF statement on the following line to determine whether the comment flag was set during dd_processing. If the comment flag is set to Y, execute the following statements in the DO loop.
4. Code the following DO loop to set each record for this statement to a comment. The DO loop sets the delete flag back to N after the statement is deleted.

```
/*********************************************************************/
/*  Raw Data Processing  Logic                                       */
/*********************************************************************/
RAW_DATA_PROCESSING:
 IF delete_flag = 'Y' Then
  Do
 $CA.RCOUNT = 0   /* Setting record count to zero deletes statement */
 delete_flag = 'N'     /* Reset flag */
End
 IF comment_flag = 'Y' Then
  Do
$CA.RCOUNT = $CA.RCOUNT + 0
   Do n = 1 to $ca.rcount
    temp_work = SUBSTR($CA.RECORD.n,3,77)
    temp_record = '//*'temp_work
    $CA.RECORD.n = temp_record
   End
 comment_flag = 'N'     /* Reset flag */
End
Return
```

JCLNeat Original JCL Report

This report is a sequential listing of the statements read from the input stream specified to JCLNeat using the REXXMEM option. All error messages issued by the JCLNeat REXX interface appear in this report and are prefixed with CAY6541.

```
   CA                        JCLNeat ORIGINAL JCL
   CA JCLCHECK             SYS: xxxx

IN  //PRODJOB JOB (40100000),'S. BELL',CLASS=O,MSGCLASS=X,              00010004
IN  //     NOTIFY=USER02,PRTY=15                                        00020004
CAY6541W Class "O" is reserved for second shift usage, retry after 5PM, it is only 11:14am class set for first shift use
IN  //*   PRODUCTION GENERAL LEDGER JOB                                 00021005
IN  //S0 EXEC PGM=GLPGM                                                00022005
IN  //OBSOLETE DD DSN=BOGUS.DSNAME,DISP=SHR                            00023005
CAY6541W DD named OBSOLETE was coded and is no longer required, this statement will be deleted
IN  //DD1 DD DSN=CAI.GL.MASTER,DISP=(NEW,CATLG,DELETE),                00023105
IN  //   UNIT=BOGUS,SPACE=(CYL,(500,50),RLSE),                         00024004
IN  // DCB=(RECFM=VB,LRECL=255,BLKSIZE=16320,DSORG=PS)                  00025004
CAY6541I Unit name coded BOGUS is no longer valid and will be removed
IN  //DD2 DD DSN=CAI.GL.RATES,DISP=(NEW,CATLG,DELETE),                 00025105
IN  //   UNIT=CART                                                              00025205
IN  //DD3 DD DSN=CAI.GL.OUT1,DISP=(NEW,CATLG,DELETE),                 00025305
IN  //   UNIT=CART                                                     00025405
IN  //DD4 DD DSN=CAI.GL.OUT2,DISP=(NEW,CATLG,DELETE),                 00025505
IN  //   UNIT=CART                                                     00025605
IN  //DD5 DD DSN=CAI.GL.MERGE,DISP=(NEW,CATLG,DELETE),                00025705
IN  //   UNIT=CART                                                     00025805
IN  //DD6 DD DSN=CAI.GL.SPLIT,DISP=(NEW,CATLG,DELETE),                00025905
IN  //   UNIT=CART                                                     00026005
IN  //COMMENT DD DSN=BOGUS.DSNAME,DISP=SHR                            00027005
CAY6541I  DD named COMMENT will be commented out, it is no longer vaild
IN  //S1 EXEC PGM=IEFBR14                                              00028006
CAY6541E Step S0 is using 5 cartridge drives, please reduce utilization
CAY6541S JOB PRODJOB_ has encountered too many site violations
```

Reformatted JCL Report

This report is a sequential listing of the JCL statements generated by JCLNeat in reformatting the JCL. Each statement line is prefixed with an identifier of OUT. A new page begins when a logical break is encountered in the input stream.

```
   CA                                  REFORMATED JCL
   CA JCLCHECK             SYS: xxxx

OUT //PRODJOB JOB (40100000),                                          00000010
OUT //       'S. BELL',                                                00000020
OUT //        CLASS=D,                                                 00000030
OUT //        MSGCLASS=X,                                              00000040
OUT //        NOTIFY=USER02,                                           00000050
OUT //        PRTY=15                                                  00000060
OUT //**-------------------------------------------------------------*00000070
OUT //**                                                             *00000080
OUT //*   PRODUCTION GENERAL LEDGER JOB                              *00000090
OUT //**                                                             *00000100
OUT //**-------------------------------------------------------------*00000110
OUT //S0   EXEC PGM=GLPGM                                            00000120
OUT //DD1  DD  DCB=(BLKSIZE=16320,DSORG=PS,LRECL=255,RECFM=VB),       00000150
OUT //         DISP=(NEW,CATLG,                                       00000160
OUT //         DELETE),                                               00000170
OUT //         DSN=CAI.GL.MASTER,                                     00000180
OUT //         SPACE=(CYL,((500,50,                                   00000190
OUT //         ),RLSE)                                                00000200
OUT //DD2  DD  DISP=(NEW,CATLG,                                       00000210
OUT //         DELETE),                                               00000220
OUT //         DSN=CAI.GL.RATES,                                      00000230
OUT //         UNIT=CART                                              00000240
OUT //DD3  DD  DISP=(NEW,CATLG,                                       00000250
OUT //         DELETE),                                               00000260
OUT //         DSN=CAI.GL.OUT1,                                       00000270
OUT //         UNIT=CART                                              00000280
OUT //DD4  DD  DISP=(NEW,CATLG,                                       00000290
OUT //         DELETE),                                               00000300
OUT //         DSN=CAI.GL.OUT2,                                       00000310
OUT //         UNIT=CART                                              00000320
OUT //DD5  DD  DISP=(NEW,CATLG,                                       00000330
OUT //         DELETE),                                               00000340
OUT //         DSN=CAI.GL.MERGE,                                      00000350
OUT //         UNIT=CART                                              00000360
OUT //DD6  DD  DISP=(NEW,CATLG,                                       00000370
OUT //         DELETE),                                               00000380
OUT //         DSN=CAI.GL.SPLIT,                                      00000390
OUT //         UNIT=CART                                              00000400
OUT //**-------------------------------------------------------------*00000410
OUT //**                                                             *00000420
OUT //*COMMENT DD DSN=BOGUS.DSNAME,DISP=SHR                          00000430
OUT //**                                                             *00000440
OUT //**-------------------------------------------------------------*00000450
```

This report reflects the changes that were made by the CAZ1NREX EXEC in this section. Specifically UNIT=BOGUS is removed, ddname OBSOLETE is removed, and ddname COMMENT is commented out of the JCL.