# CA JCLCheck™ Workload Automation

## Best Practices Guide

**Version 12.0.00**

# CA Technologies Product References

This document references the following CA Technologies products:

- CA Chorus™ Software Manager (CA MSM)
- CA Workload Automation CA 7® Edition (CA WA CA 7 Edition)
- CA Scheduler® Job Management (CA Scheduler)
- CA Workload Automation ESP Edition (CA WA ESP Edition)
- CA APCDOC™ Automated Job Documentation (CA APCDOC)
- CA ACF2™ (CA ACF2)
- CA Top Secret® (CA Top Secret)
- CA Roscoe® Interactive Environment (CA Roscoe)
- CA InterTest™ Batch (CA InterTest Batch)

# 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.

**Best Practices Guide Process**

These best practices are based on customer experience reported through interviews with development, technical support, and technical services. Therefore, many of these best practices are a collaborative effort stemming from customer feedback.

To continue to build on this process, we encourage you to share common themes of product use that might benefit other users. Please consider sharing your best practices with us.

To share your best *practices, c*ontact us at techpubs@ca.com and preface your email subject line with "Best Practices for product name" so that we can easily identify and categorize them.

# Contents

# Chapter 1: Introduction

This section contains the following topics:



## Purpose of this Guide

The guide provides a brief introduction to the CA Technologies mainframe management strategy and features, and describes the best practices for installing and configuring CA JCLCheck.

## Audience

The intended audience of this guide is systems programmers and administrators who install, configure, deploy, and maintain CA JCLCheck.

## Mainframe 2.0 Overview

Mainframe 2.0 is our strategy for providing leadership in the mainframe operating environment. We intend to lead the mainframe marketplace for customer experience, Out-Tasking solutions, and solution innovation. After listening to customer needs and requirements to keep the mainframe operating environment viable and cost-effective, we are providing new tools to simplify usage and to energize this operating environment for years to come.

CA Mainframe Software Manager™ (CA MSM) is an important step in realizing the Mainframe 2.0 strategy. CA MSM simplifies and standardizes the delivery, installation, and maintenance of mainframe products on z/OS systems. CA MSM has a web-based interface with a modern look and feel for managing those solutions. As products adopt Mainframe 2.0 features and CA MSM services, you can acquire, install, and manage your software in a common way.

We follow the IBM z/OS packaging standards using SMP/E, with some additional CA Technologies qualities of service added, to make installation simple and consistent. Additionally, through the synchronization of product releases and the use of common test environments, we will declare a yearly mainframe software stack that includes many new releases with enhanced functionality. This stack is certified for interoperability across the CA Technologies mainframe product portfolio and the base IBM z/OS product stack.

# Mainframe 2.0 Features

Mainframe 2.0 has the following main features:

**CA Mainframe Software Manager (CA MSM)**

Delivers simplified acquisition, installation, and deployment capabilities using a common z/OS-based web application delivered through a browser-based UI. CA MSM includes the following services:

**Product Acquisition Service (PAS)**

Facilitates the acquisition of our mainframe products and services, including product base installation packages and program temporary fixes (PTFs). This service integrates the inventory of products available on your system with CA Support, providing a seamless environment for managing and downloading software and fixes onto your system.

**Software Installation Service (SIS)**

Facilitates the installation and maintenance of our mainframe products in the software inventory of the driving system. This service enables you to browse and manage the software inventory using a web interface, and automates tasks for products that use SMP/E to manage installation. You can browse downloaded software packages, and browse and manage one or more consolidated software inventories (CSIs) on the driving system.

**Software Deployment Service (SDS)**

Facilitates the deployment of CA Technologies mainframe products from the software inventory of the driving system. This service enables you to deploy installed products that are policy-driven with a set of appropriate transport mechanisms across a known topology. The enterprise system topology can include shared DASD environments, networked environments, and z/OS systems. Policies represent a combination of metadata input and user-supplied input. Metadata input identifies the component parts of a product. User-supplied input identifies the deployment criteria, such as where it goes and what it is named.

**Software Configuration Service (SCS)**

Facilitates the mainframe products configuration from the software inventory of the driving system to the targeted z/OS mainframe operating system. SCS guides you through the configuration creation process, and through the manual steps to implement the configuration. In addition, SCS includes an address space communications service running on each targeted z/OS system.

**Electronic Software Delivery (ESD)**

Enables you to get our products from an FTP server. We have improved this process so that you no longer need to build a tape to install the product.

**Best Practices Management**

Integrates with IBM Health Checker for z/OS to verify that deployed software follows our best practices. The health checks continually monitor the system and software to provide feedback on whether the software continues to be configured optimally.

**Best Practices Guide**

Provides best practices for product installation and configuration.

# Chapter 2: Installation and Configuration Best Practices

CA JCLCheck represents critical software in support of meeting your production processing requirements as it minimizes or eliminates JCL problems in your environment. This guide represents the best practices that are associated with installation of the software addressing SMP/E considerations, APF authorization, security, and user modifications.

**Business Value:**

To ensure a successful, quicker, and more accurate installation of CA JCLCheck, follow the installation and configuration best practices.

**Note:** Review the *Installation Guide* and the *Programming Guide* before installing a new version of CA JCLCheck. If you are upgrading to a new release, see the *Release Notes.* The Release Notes help you determine the impact of changes with the new release.

This section contains the following topics:



## CA JCLCheck and CA JCLCheck Common Component

CA JCLCheck is the full product. CA JCLCheck Common Component is a subset of the CA JCLCheck product and has limited functions.

CA JCLCheck provides interfaces to validate JCL from TSO/ISPF, and from a batch job. It also includes the JCLNeat feature for reformatting and making mass changes to the JCL. CA JCLCheck produces reports after the JCL is validated.

CA JCLCheck Common Component is shipped free of charge with several CA Technologies products. CA JCLCheck Common Component can only be executed from within the primary CA Technologies product. It does not have the ISPF, batch, and JCLNeat interfaces or the reporting function.

# SMP/E Data sets

For ease of installation and maintenance, install CA JCLCheck in its own SMP data sets. When installing a new version or release of CA JCLCheck, use a new TARGET and DISTRIBUTION zone to prevent the previous version or release from getting deleted.

Do not install CA JCLCheck Common Component if you already have the full blown CA JCLCheck. Installing both products can create problems of mixed modules when either product is invoked.

If you do not have CA JCLCheck but receive numerous copies of the CA JCLCheck Common Component from several of your primary CA Technologies products, install only one version in its own CSI. All products can share these libraries.

The following list shows CA Technologies products that use the CA JCLCheck Common Component. If the Product Release shows only the release level, this product delivers CA JCLCheck Common Component as a separate installable file. If the Product Release contains the word 'included', the JCLCheck Common Component modules are delivered with the primary product file. The JCLCheck Common Component modules are automatically installed when the primary product is installed.

| Product | Release | Release |
|---|---|---|
| CA Workload Automation CA 7 Edition | 11.1/11.3 | |
| CA APCDOC | 1.3 | |
| CA Deliver | 11.5 | 1.7 (included) |
| CA Intertest Batch | 8.5 | 8.0 (included) |
| CA Roscoe IE | 6.0 SP09 | 6.0 SP08 (included) |
| CA Scheduler JM | 11.0 | |
| CA Workload Automation ESP Edition | 11.3 | |

# APF Authorization

The CA JCLCheck CAZ2LOAD and CD51LOAD libraries must be APF-authorized. If the CAZ2LOAD is not APF-authorized, unpredictable results such as ABENDs or incorrect messages can occur.

# Security

CA JCLCheck must be given READ access to all input sources, such as JCL libraries, procedure libraries, utility control members, and catalogs. Access to existing partitioned data sets that reference a member name in the JCL must also be given access to validate the JCL. When using the AUTOPROC feature, security is called for each PROCLIB that JES2 returns. If security fails, CA JCLCheck issues message CAY6488, and the library is omitted from allocation.

CA JCLCheck can also interface with external security products such as CA ACF2, CA Top Secret, and RACF. CA JCLCheck verifies whether a user has access authority to DATA SET, DASDVOL, PROGRAM, STORCLAS, and MGMTCLAS resources. This type of security prevalidation helps reduce the incidence of S913 ABENDs and job failures when the job is submitted. The security prevalidation option is activated by using runtime options SECURITY(options) and USER(userID). The runtime option USER(userID) is optional. USER(userID) is required only when you are checking if the SECID is different from the SECID of the submitter.

# User Modification Tables

CA JCLCheck has tables (CSECT) that are user-modifiable through predistributed user modifications (USERMODs). For a list of user-modifiable tables, see (Optional) Install CA JCLCheck Tables for Modification in the *Installation Guide*. You execute the USERMOD that assembles and links the table into the CA JCLCheck load modules. These USERMODs are stored in the CA JCLCheck SAMPJCL library and their member names have an "MZ" prefix.

USERMODs must be reinstalled after each CA JCLCheck version or release upgrade. Normal maintenance (PTFs) generally does not require a reinstall of the USERMOD. When a PTF requires a reinstall of the USERMOD, you are informed during the SMP APPLY step of the PTF. The following message is an example of what you receive:

```
GIM38201E ** THERE IS A MODID ERROR FOR SRC ENTRY JCLMSG IN SYSMOD RO07380.
GIM31902I    SYSMOD RO07380 DOES NOT SPECIFY MZ2B023 ON THE PRE OR SUP OPERAND.
```

In this example, the corrective action is to RESTORE USERMOD MZ2C023, re-APPLY PTF RO07380, then re-APPLY USERMOD MZ2C023.

The following examples show how the CA JCLCheck tables are modified.

### Example: Set Default Runtime Options

This example shows how to set the default runtime options to "CTL FULL JOB LIST XREF PXREF(RPT) PROCXREF SXREF AU FLUSHRC(00) MCOSYS(CAI.JCK.MSGOPT)".

The default runtime options are defined in the JCLDFLT CSECT, which is stored in the CAZ2OPTN library. This library is SMP controlled and must not be directly modified. Instead, place the desired options in USERMOD MZ1C024 as shown, and submit the USERMOD for execution. Be sure to validate the sequence numbers in column 73-80.

```
…
./ CHANGE NAME=JCLDFLT
   CAZ2BOX TITLE='JCLDFLT - DEFAULT OPTIONS CSECT'
OPT     DC      C'CTL FULL JOB LIST XREF PXREF(RPT) PROCXREF SXREF  '
        DC      C'AU FLUSHRC(00)  '
        DC      C'MCOSYS(CAI.JCK.MSGOPT)  '
./ ENDUP
…
```

### Example: Remove Requirements

This example removes the requirements for the programmer name and accounting information fields on the JOB statement.

The JCL decoding tables are defined in member MVS4XTBL, which is stored in the CAZ2OPTN library. To modify the decoding tables, edit USERMOD MZ2C012 from the SAMPJCL library as shown, and submit the USERMOD for execution. Be sure to validate the sequence numbers in column 73-80.

```
…
./ CHANGE NAME=MVS4XTBL
JCLTABLE CAZ2BOX HEAD='DECODING TABLES FOR JCLCHECK - ',
    $J6PCB NAME=ACCOUNT,PTB=JOBACTG,REQ=NO,
    $J6PCB NAME=PRGRAMMR,VTB=JPRGMMR,REQ=NO,
./ ENDUP
…
```

# Using the EDCHEK Interface

The EDCHEK feature is the most commonly used ISPF interface and is invoked to validate JCL while in an ISPF Edit session. To use EDCHEK, execute either command !JCK or EJCK. These commands validate the JCL and display the results on the terminal. The command !JCK executes a program while EJCK executes a CLIST, which allows the passing of runtime options. For example, the command ” EJCK LIST” passes the option LIST to display all the reports requested. The command “EJCK PANEL” displays an options panel before executing.

**To use the EDCHEK interface:**

1. Place the CA JCLCheck CAZ2LOAD in the LINKLIST, as best practice. The CA JCLCheck CD51LOAD must also be in the LINKLIST when ChekPlex is used. If these libraries cannot go in the LINKLIST, then they must be in the STEPLIB allocation in the TSO logon procedure.

   **Note:** All the STEPLIB concatenated libraries must be APF-authorized.

2. Allocate the following libraries to your logon procedure:

   - CAZ2CLS0 library to SYSPROC concatenation
   - CAZ2PNL0 library to ISPPLIB concatenation
   - CAZ2SKLO library to ISPSLIB concatenation
   - CAZ2MSG0 library to ISPMLIB concatenation

   Alternately, you can use the LIBDEF method of allocating your system libraries as follows:

   - ISPEXEC LIBDEF ISPMLIB DATASET ID('CAI.CAZ2MSG0')
   - ISPEXEC LIBDEF ISPPLIB DATASET ID('CAI.CAZ2PNL0')
   - ISPEXEC LIBDEF ISPSLIB DATASET ID('CAI.CAZ2SKL0')
   - ISPEXEC LIBDEF ISPTLIB DATASET ID('CAI.CAZ2TBL0')

     **Important:** Do not use LIBDEF ISPLLIB to allocate the CA JCLCheck load libraries (CAZ2LOAD and CD51LOAD). ISPLLIB modules cannot run authorized.

3. Add the following entries to SYS1.PARMLIB(IKJTSO*nn*), as APF-authorization for CA JCLCheck in TSO/ISPF mode is required:

   - CAZ2CTSO in the AUTHTSF table
   - JCLCHECK in the AUTHCMD table
   - JCLCHECK in the AUTHPGM table

     To validate this setup, enter TSO CAZ1TSF. When APF-authorized, the display shows:

     CAZ1TSF - JCLCHECK AUTHTSF ENTRY 'CAZ2CTSO' FOUND

     CAZ1TSF - JCLCHECK AUTHCMD ENTRY 'JCLCHECK' FOUND

     CAZ1TSF - JCLCHECK AUTHPGM ENTRY 'JCLCHECK' FOUND

4. Allocate the EDCHKDD DD data set to your TSO logon procedure or logon CLIST.

   The EDCHKDD DD statement allocates a data set that contains CA JCLCheck runtime options.

# Using the ISPF Panel Interface

The CA JCLCheck ISPF panels interface lets you perform many functions ranging from a single JCL check to an entire PDS, or batch mode processing. The panels also include a JCLNeat option to reformat your JCL to your shop standards.

**To use the ISPF panels interface:**

1. Place the CA JCLCheck CAZ2LOAD in the LINKLIST, as best practice. The CA JCLCheck CD51LOAD must also be in the LINKLIST when ChekPlex is used. If these libraries cannot go in the LINKLIST, then they must be in the STEPLIB allocation in the TSO logon procedure.

   **Note:** All the STEPLIB concatenated libraries must be APF-authorized.

2. Allocate the following libraries to your logon procedure:

   - CAZ2CLS0 library to SYSPROC concatenation
   - CAZ2PNL0 library to ISPPLIB concatenation
   - CAZ2SKLO library to ISPSLIB concatenation
   - CAZ2MSG0 library to ISPMLIB concatenation

   Alternately, you can use the LIBDEF method of allocating your system libraries as follows:

   - ISPEXEC LIBDEF ISPMLIB DATASET ID('CAI.CAZ2MSG0')
   - ISPEXEC LIBDEF ISPPLIB DATASET ID('CAI.CAZ2PNL0')
   - ISPEXEC LIBDEF ISPSLIB DATASET ID('CAI.CAZ2SKL0')
   - ISPEXEC LIBDEF ISPTLIB DATASET ID('CAI.CAZ2TBL0')

     **Important:** Do not use LIBDEF ISPLLIB to allocate the CA JCLCheck load libraries (CAZ2LOAD and CD51LOAD). ISPLLIB programs cannot run authorized.

3. Add the following entries to SYS1.PARMLIB(IKJTSO*nn*), as APF-authorization for CA JCLCheck in TSO/ISPF mode is required:

   - CAZ2CTSO in the AUTHTSF table
   - JCLCHECK in the AUTHCMD table
   - JCLCHECK in the AUTHPGM table

# Using the ChekPlex Feature

The ChekPlex feature enables you to issue a CA JCLCheck validation request from one system and the actual validation occurs on another system. ChekPlex uses the CA General Transaction Server (GTS) common component to communicate and to route JCL, procedures, and results between systems.

**To use ChekPlex, follow these steps:**

1. Install CA GTS.

   CA GTS is included in the CA JCLCheck product tape. CA GTS is installed by selecting FMID CD51110 during the RECEIVE, APPLY, and ACCEPT of CA JCLCheck. The CA GTS loadlib (CD51LOAD) must be a PDSE data set. Exclude this step if CA GTS is already installed with another CA Technologies product.

2. Configure CA GTS according to the *CA General Transaction Server User Guide.* Additionally, see The ChekPlex Feature in the CA JCLCheck *Programming Guide.*

3. Activate the GTS servers.

   ChekPlex uses the following server nodes:

   **LOCAL**

   Where the request for JCL validation originates.

   **TARGET**

   Where the JCL is validated.

   **SOURCE**

   Where the JCL resides.

   In most cases LOCAL and SOURCE are the same server.

   The CA GTS started task must run on each participating node.

4. Add the CA JCLCheck loadlib (CAZ2LOAD) and the CA GTS loadlib (CD51LOAD) to a STEPLIB DD or to the LINKLIST.

5. Create a REMVAL data set and allocate this data set to a REMVAL DD.

   The REMVAL data set contains the Remote Validation options. For a description of these options, see The ChekPlex Feature in the *Programming Guide*.

6. Set the REMOTE runtime option.

   See the following sample JCLCheck job:

```
// JOB STATEMENT
//STEP1 EXEC PGM=JCLCHECK,PARM='OPTIONS(JCLOPTS)'
//STEPLIB  DD DISP=SHR,DSN=cai.jclcheck.caiload
//         DD DISP=SHR,DSN=cai.gts.caipld
//SYSTERM  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSPROC  DD DSN=SYS1.PROCLIB,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSGRAPH DD SYSOUT=*,DCB=RECFM=FBA
//SYSIN DD DSN=cai.test.jcllib,DISP=SHR <= JCL lib containing jcl01
//JCLOPTS  DD *
REMOTE PDS INCLUDE(jcl01)
//REMVAL DD *
LOCAL gtscpsya  <= GTS server name
SOURCE gtscpsya <= GTS server name
TARGET gtscpsyb <= GTS server name
/*
```

# The JCLNeat Component

JCLNeat is included with CA JCLCheck and is used to reformat JCL to conform to your shop standards. The JCL reformatting is done by using processing features, formatting options, and a table-driven keyword ordering. For example, JCL can be reformatted to align in certain columns. All JCL keyword parameters can be put in a sequence order, such as, DSN, DISP, UNIT, SPACE, and DCB respectively.

In addition, global changes can be made to JCL. For example, you can change UNIT=3380 to UNIT=SYSDA in all procedure libraries. You can use the JCLNeat Global String replacement feature for this request with the following control statements:

```
//GSRCNTL DD *
|UNIT=3380|REPL|
|UNIT=SYSDA|
```

The JCLNeat REXX interface is also available for changes that require logic. You can invoke JCLNeat through the ISPF panel, ISPF Edit, and in batch mode. Input to JCLNeat can be one JCL, a sequential file, or an entire PDS library. JCLNeat is not used for JCL validation and it expects the input JCL to be error-free. As best practice, the JCL should be processed through CA JCLCheck before and after the JCLNeat changes.

# Chapter 3: Customization

CA JCLCheck presents significant opportunities for customization to meet the needs of your environment and users. To optimize the product usage at your site, use the following best practices.

**Business Value**

CA JCLCheck provides many interfaces such as ISPF, TSO Foreground, and batch processing. There are many options that affect the results regardless of how CA JCLCheck is invoked. Best practices for Runtime Options, EDCHEK, ISPF Panels, AUTOPROC, reporting, message handling, and REXX are included to maximize your use as required.

This section contains the following topics:



## Default Runtime Options

Default runtime options are customized by each site. As best practice, the default options should consist of runtime options that are used no matter how CA JCLCheck is invoked. For instance, if REXX is used to enforce site standards, then the STDREXX option should be in the default option.

To get a current list of the default options, execute CA JCLCheck in batch mode and include the runtime option "LIST". In the execution listing, the default options are displayed under DEFAULT PARAMETERS as follows:

```
DEFAULT PARAMETERS: AU CTL FULL JOB LIST XREF PXREF(RPT) PROCXREF SXREF
```

To modify the default runtime options, apply CA JCLCheck USERMOD MZ1C024 from the SAMPJCL data set. Validate the sequence numbers in column 73-80.

```
...
./ CHANGE NAME=JCLDFLT
   CAZ2BOX TITLE='JCLDFLT - DEFAULT OPTIONS CSECT'
OPT    DC    C'CTL FULL JOB LIST XREF PXREF(RPT) PROCXREF SXREF'
       DC    C'AU FLUSHRC(00) STDREXX(YOURREXX)'
./ ENDUP
```

# EDCHEK Options

As best practice, allocate an EDCHKDD data set to the TSO/ISPF session either through the TSO logon procedure or CLIST. The EDCHKDD data set contains the runtime options EDCHEK (command !JCK or EJCK) uses from a TSO/ISPF Edit session.

The following EDCHKDD statement is a sample in the TSO logon proc:

```
//EDCHKDD DD DSN=CAI.CAIOPTN(CAZ1SEDA),DISP=SHR
```

Several examples are in data set CAI.CAZ2OPTN(CAZ1SEDx). The most commonly used member is CAZ1SEDA, which contains the AUTOPROC option:

```
EDCHEK TSOE    -
         AU
```

Additionally, you can add options to override your default options. In the following example, option NOLIST prevents the wrap around output, and option EDCHEL expands procedures:

```
EDCHEK TSOE NOLIST NOSXREF EDCHEL -
         AU
```

If you are not using AUTOPROC (NOAU), then CAZ2OPTN(CAZ1SEDP) can be used and the procedure libraries must be manually allocated as follows:

```
EDCHEK PROC(PROC00)
ALLOC FI(PROC00) DA(SYS1.PROCLIB)
ALLOC FI(*)      DA(SYS2.PROCLIB)
ALLOC FI(*)      DA(USER.PROCLIB)
ALLOC FI(PROC01) DA(USER2.PROCLIB)
GO
```

The CA JCLCheck runtime options are derived from different sources and are processed in the following order:

1. JCLDFLT module out of the box or modified by CA JCLCheck USERMOD MZ1C024.
2. EDCHKDD DD data set that is allocated to the TSO session.
3. EDCHEK options panel (ID JCK21) displayed by entering command %EJCK PANEL.

The list of options in effect, that always display in the CA JCLCheck report, reflect the runtime options used.

# ISPF Panel Options

The CA JCLCheck ISPF panels interface uses runtime options that come from several sources. The runtime options are processed in the following order:

1. Module JCLDFLT out of the box, or modified by CA JCLCheck USERMOD MZ1C024.
2. CA JCLCheck ISPF Options Specification menus. These menus are displayed by selecting option 0 from the CA JCLCheck primary menu.
3. Any runtime option can be added or overridden on the CA JCLCheck Foreground Invocation and Batch Submit panels. These panels are both JCK0201.
4. CLIST JCKUSR (AUTOPROC option only).

   The runtime options in the CA JCLCheck ISPF panels interface can be customized for each individual user.

# AUTOPROC Option

The AUTOPROC runtime option enables CA JCLCheck to extract the current procedure library configuration (PROCLIB) from the JES2 or JES3 address space. Therefore, you do not have to manually allocate your proclibs when running CA JCLCheck. As best practice, enable the AUTOPROC option.

Instances occur when you must search a user procedure library before the system procedure libraries. You can accomplish this task by using the CA JCLCheck runtime option OPROC(ddname).

Before you run CA JCLCheck, specify the DD name on the OPROC option. The DD name must define the procedure libraries that you want to search ahead of the system procedure libraries. If executing CA JCLCheck from ISPF (using the EDCHEK or the ISPF panels), the OPROC DD name must be preallocated. If defining overriding procedure libraries to a DD name is not feasible, runtime option OPROCLIB(library) can be used instead. The OPROCLIB option is mutually exclusive with OPROC. Multiple OPROCLIB(library) statements can be specified, and the library that is defined to OPROCLIB must exist on the system where CA JCLCheck runs.

# Report Options

When JCL is validated, 13 different types of reports can be produced and written to the SYSPRINT data set. The CA JCLCheck runtime options control the number and types of reports.

The following list contains the types of reports that are produced and the associated runtime options. As best practice, review the need for certain reports at your site to reduce overhead especially when the entire JCL library is checked.

**Report 0 - LISTING OF OPTIONS**

Use the LIST runtime option. This report displays the default runtime options, override options, and runtime OPTIONS in EFFECT.

**Report 1 - LISTING OF JOBSTREAM JCL**

Use the JOB runtime option. This report is a card image listing of the JCL submitted to CA JCLCheck for validation.

**Report 2 - LISTING OF MERGED JCL**

Use the FULLLIST runtime option. This report is the entire JCL including expanded procedure statements.

**Report 3 - DATA SET CROSS REFERENCE**

Use the XREF runtime option. This report lists all data sets and PDS members that the current job being validated is using.

**Report 4 - PROGRAM CROSS REFERENCE**

Use the PXREF runtime option. This report lists all the programs that the current job being validated is using and the libraries where the programs reside.

**Report 5 - REPORT REPORT**

Use the RPTRPT runtime option. This report lists any statement specifying the SYSOUT parameter.

**Report 6 - ERROR MESSAGES**

Use the ERROR runtime option. This report lists all CA JCLCheck error messages that are detected for the current job being validated.

**Report 7 - SUMMARY DATA SET CROSS REFERENCE**

Use only the XREF and SXREF runtime options. To produce Report 3, Report 4, Report 8, and Report 7, use the XREF and SXREF (RPT) runtime options. This report lists all the data sets and members that all the jobs being submitted are using, for this CA JCLCheck run.

**Report 8 - SUMMARY PROGRAM CROSS REFERENCE**

Use only the PXREF and SXREF runtime options. To produce Report 3, Report 4, Report 7, and Report 8, use the PXREF and SXREF (RPT) runtime options. This report lists all the programs that all the jobs being submitted are using, for this CA JCLCheck run.

**Report 9 - SUMMARY REPORT REPORT**

Use the RPTRPT and SXREF runtime options. This report lists all SYSOUT statements from all the jobs being submitted for this CA JCLCheck run.

**Report 10 - FLOW DIAGRAM**

Use the GRAPH runtime option. This report is a graphic representation of the flow of the JCL being validated. Report 10 is displayed to a SYSGRAPH DD data set.

**Report 11 - PROCEDURE CROSS REFERENCE**

Use the PROCXREF runtime option. This report lists all procedures used in the validation of JCL. This report also lists a cross reference of all symbolic parameters used in the procedures and the libraries where the procedures are found.

**Report 12 - JCL PROCESSING AUDIT REPORT**

Use the AUDIT runtime option. This report shows a statistical summary, input records, and averages of jobs, steps, and DD statements.

# Changing and Suppressing Message Severity Level

The severity of the CA JCLCheck error messages can be altered or suppressed. You can do this task with the following methods:

1. Change table JCLMSG by applying a USERMOD.
2. Internally by the program which issued the error message.
3. Use the MCOSYS runtime option.
4. Use the MCOUSR runtime option.
5. Code a REXX exec.
6. Code an Assembler user error exit.

CA Technologies recommends using the runtime options MCOSYS or MCOUSR due to ease of use and quicker implementation.

The MCOSYS runtime option specifies the data set name that points to the system message control option file. The data set must be a sequential file or a PDS member with a logical record length of 80.

The following syntax is for the MCOSYS option:

```
MCOSYS(sequential.file.name) | MCOSYS(pds.file.name,member)
```

The following format is the format of the message control file:

```
Column  1-3   Indicates the valid CA JCLCheck WA message number from 001-999 (CAY6nnn)
Column    4      Must contain a comma
Column   5-6     Indicates the message severity
          -1      Suppress the message
         00-03    Informational severity
         04-07    Warning severity
         08-11    Error severity
         12-15    Serious error severity
Column   7-80    Comments
```

Example:

```
027,04      CHANGE #27 $DMSG 'PROCEDURE','NOT FOUND'  TO SEV 4
079,-1      SUPPRESS #79 $DMSG 'DATA SET','SPECIFIED AS OLD OR SHR
```

**Note:** If the line starts with an asterisk (*) in column 1, it is considered to be a comment.

Entries can be placed in any sequence and duplicate entries override previous entries. As a best practice, keep the message numbers in sequence to simplify locating and entry.

The MCOUSR runtime option enables you to specify the DDname of a user message control option file. The DDname must be allocated in your TSO logon proc or to your TSO/ISPF session. When running CA JCLCheck in batch mode, the DDname must also be defined in the CA JCLCheck job. The format of the MCOUSR control file is identical to the MCOSYS control file. If MCOUSR is used, MCOUSR takes precedence over the MCOSYS control file.

The following syntax is for the MCOUSR option:

```
MCOUSR[ddname]
```

The DDname is an optional parameter and if it is omitted, MCOUSR is used as the DDname:

```
//MCOUSR  DD  DISP=SHR,DSN=DATASET1.NAME
```

The sequence of processing the message control is in the following order:

1. JCLMSG is processed first.
2. The program that issued the error message.
3. The MCOSYS runtime option.
4. The MCOUSR runtime option.
5. REXX message processing.
6. Assembler user error exit (CAZ1XERR).

   This order of precedence means that the CAZ1XERR user exit is processed last.

The use of MCOSYS or MCOUSR adds flexibility and lets you quickly change the message severity.

# REXX Interface for CA JCLCheck

CA JCLCheck provides a REXX interface to help enforce site standards by issuing messages when the standards are violated. As best practice, use the REXX interface instead of the Assembler user exit. The REXX exec is easier to implement and does not require a USERMOD to assemble and link it into CA JCLCheck. A sample REXX exec named CAZ1REXX is provided in the CAZ2OPTN library. Use CAZ1REXX as a template to create your own standards. The REXX interface for CA JCLCheck cannot be used to alter JCL statements.

Before you code your REXX exec, see REXX for JCLCheck and JCLNeat in the *Programming Guide.*

To activate the REXX interface for CA JCLCheck, the following tasks must be done:

- Allocate the library containing the REXX exec to the SYSEXEC DD statement. The SYSEXEC DD statement must be in the CA JCLCheck job (batch mode) or TSO logon proc/CLIST (ISPF mode).
- Add runtime option STDREXX (*name*), where *name* is the REXX exec found in the //SYSEXEC DD statement.
- Add runtime option REXXMSG if the Message Processing routine is used.

# REXX Interface for JCLNeat

The JCLNeat REXX interface is provided to help enforce site standards by issuing messages when the standards are violated and updating the JCL according to site standards. A sample REXX exec named CAZ1NREX is provided in the CAZ2OPTN library. Use CAZ1NREX as a template to create your own standards.

Before you code your REXX exec, see REXX for JCLCheck and JCLNeat in the *Programming Guide.*

To activate the REXX interface for JCLNeat, the following tasks must be done:

- Allocate the library containing the REXX exec to the SYSEXEC DD statement. The SYSEXEC DD statement must be in the JCLCheck job (batch mode) or TSO logon proc/CLIST (ISPF mode).
- Add the runtime option REXXMEM=*name,* where name is the REXX exec found in the //SYSEXEC DD statement.

# Chapter 4: Maintenance

CA JCLCheck maintenance is provided from the following sources:

- The Download Center - Published Solutions on the CA Support web site.
- The CA Recommended Service (CA RS) file. CA RS is released quarterly and is not cumulative. CA RS is installed only from CA MSM.
- Aggregate Maintenance (AM) ESD PAX file that is distributed as a PTF. AM is released on an as-needed basis and is not cumulative. AM is installed outside of CA MSM.

**Business Value:**

Application of regular maintenance is critical to minimize product problems.

As a best practice, keep CA JCLCheck current with all published solutions applied using CA MSM. Visit the CA JCLCheck product home page frequently for Product News, and check the Maintenance Grid under the Product Status section. If CA MSM is not used to apply maintenance, then the Maintenance Grid is useful. The Maintenance Grid contains links to the Aggregate Maintenance fix, fixes since the last Aggregrate Maintenance, the SMP/E Error HOLDDATA bucket, HIPERs, and product documentation changes.

When downloading solutions from support.ca.com, both component name "CA JCLCheck" and "CA JCLCheck Common Component" must be selected. When a PTF has a ++HOLD action, be sure to review it thoroughly before applying the PTF. As best practice, always run APPLY CHECK first and never force BYPASS (ID) on any PTFs. Also, PTFs can be ACCEPTed, but APARs (test fix) and USERMODs must never be ACCEPTed.

CA JCLCheck does not require an IPL after installing PTFs. If a PTF was for the SUBCHEK and/or JESCHEK modules, you may need to execute CAIRIM. If CA JCLCheck is in the LINKLIST, the LINKLIST must be refreshed.

# Chapter 5: Troubleshooting

Product problems occur and by following the best practices for Troubleshooting, helps to ensure a more successful support experience and quicker problem resolution.

**Business Value:**

Providing CA Support with the proper documentation helps in a quicker diagnosis and resolution to problems reported.

This section contains the following topics:



## Sample JCL for Batch Output

To help speed up problem diagnostic, CA Support may request the CA JCLCheck errors be reproduced using CA JCLCheck in batch mode (EXEC PGM=JCLCHECK). The use of batch and a STEPLIB DD to the CA JCLCheck load library eliminates other factors.

The following JCL executes proc CAZ1JCHK from the CA JCLCheck CAZ2PROC data set:

```
//Z1EXJCHK  JOB ...
//Z1EXJCHK  EXEC CAZ1JCHK, OPTION='O(OPTS)
//STEPLIB   DD  DISP=SHR,DSN=CAI.JCLCHECK.CAILOAD   <== change
//SYSIN     DD  DISP=SHR,DSN=YOUR.JCLLIB(JOBX)      <== change
//OPTS      DD  *
AU JOB FULL LIST XREF SXREF PXREF
/*
```

**Usage notes:**

- Remove AU if you are not using AUTOPROC.
- More sample JCL is located in the SAMPJCL data set, member Z1EXJCHK.

## DEBUG Option

CA Support may request an SVCDUMP to diagnose why a CAY6*nnnx* error message is produced. The DEBUG option takes an SVCDUMP when the error message is produced. The SVCDUMP is written to your system dump data set.

Add to the list of runtime options:

```
DEBUG(nnn,01) NOSPIE
```

### Example: Request an SVCDUMP

This example shows how to request an SVCDUMP after the third occurrence of message CAY6085W:

```
DEBUG(085,03)
```

# SYSMDUMP

CA Support may request a dump to diagnose the problem when CA JCLCheck ABENDs. To obtain a SYSMDUMP, use the following procedures:

1. Allocate a SYSMDUMP DD in the JCL of the failing job and ensure all SYSUDUMP and SYSABEND DD's are removed. Also, disable all dump analysis software for this job.

   ```
   //SYSMDUMP DD DSN=HLQ.SYSMDUMP,DISP=(MOD,CATLG),
   //            UNIT=SYSDA,SPACE=(CYL,(100,100),RLSE),
   //            DCB=(RECFM=FBS,LRECL=4160,BLKSIZE=24960)
   ```

2. Issue the following z/OS console command:

   ```
   'CD SET,SYSMDUMP=ALL'
   ```

3. Run the failing job with an additional runtime option of “NOSPIE”.
4. Terse the dump and FTP it in binary mode to CA.
5. FTP information:

   * FTP server: supportftp.ca.com
   * FTP Directory: /*xxxxxxx*/*nnnnnnnn*-01/files_from_customer

     where *xxxxxx* is the site ID and *nnnnnnnn* is the case number
   * Use your support.ca.com logon ID and password to access the FTP server.

# SVCDUMP

CA Support may request a SLIP dump when the appropriate documentation is required.

The following SLIP command triggers a dump when an 0C*x* (0C1, 0C3, 0C4, ...) ABEND occurs. CA Support can provide different kinds of SLIP instructions to gather documentation.

```
/SLIP SET,COMP=0Cx,ACTION=SVCD,ID=EDCK
```

Reply to IEE726D ENTER ADDITIONAL SLIP PARAMETERS, 'END'

```
/R __,JOBLIST=(xxxxxxxx,zzzzzzzz) or JOBNAME=xxxxxxxx
```

Reply to IEE726D ENTER ADDITIONAL SLIP PARAMETERS, 'END'

```
/R __,SDATA=(CSA,LPA,LSQA,PSA,RGN,SQA,SWA,SUM,TRT,WLM,NUC),END
```