

# CA JCLCheck™ Workload Automation

## Release Notes

Version 12.0.00



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2013 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

## CA Technologies Product References

This document references the following CA products:

- CA Chorus™ Software Manager (CA MSM)
- CA JCLCheck™ Workload Automation

## 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](mailto: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>.



# Contents

---

<b>Chapter 1: New Features</b>	<b>7</b>
Installation, Deployment, and Configuration Using CA MSM .....	7
Mainframe 2.0 Compliance .....	8
External Variable Resolution .....	8
DFSORT and ADRDSSU Utility Control Card Support.....	9
Extended Address Volume Support .....	10
JCLNeat—New REXX Variables .....	10
JCLNeat—Additional REXX Calls and Variables .....	11
JCLNeat—Global String Replacement by Statement Type .....	12
JCLNeat—Options by Statement Type .....	12
JCLNeat—REXX Dynamic Option Changes.....	13
JCLNeat—REXX Insert and Delete JCL Statements .....	13
JCLNeat PACK Option—Maximum Keywords per Line.....	14
ISPF List of REXX Rules.....	14
REXX Enhancements.....	15
Add Inactive Option Counterpart to Runtime Options .....	16
Missing DSN Bypass and Selective RUNTIME Checks .....	16
Control Card Override .....	17
Alternate User ID.....	17
Scenario Documentation.....	18
<b>Chapter 2: Enhancements to Existing Features</b>	<b>19</b>
CA MSM.....	19
<b>Appendix A: Known Issues</b>	<b>21</b>
Published Fixes for Version 12.0.00 .....	21
<b>Appendix B: Third-party Acknowledgements</b>	<b>23</b>



# Chapter 1: New Features

---

This section contains the following topics:

- [Installation, Deployment, and Configuration Using CA MSM](#) (see page 7)
- [Mainframe 2.0 Compliance](#) (see page 8)
- [External Variable Resolution](#) (see page 8)
- [DFSORT and ADRDSSU Utility Control Card Support](#) (see page 9)
- [Extended Address Volume Support](#) (see page 10)
- [JCLNeat—New REXX Variables](#) (see page 10)
- [JCLNeat—Additional REXX Calls and Variables](#) (see page 11)
- [JCLNeat—Global String Replacement by Statement Type](#) (see page 12)
- [JCLNeat—Options by Statement Type](#) (see page 12)
- [JCLNeat—REXX Dynamic Option Changes](#) (see page 13)
- [JCLNeat—REXX Insert and Delete JCL Statements](#) (see page 13)
- [JCLNeat PACK Option—Maximum Keywords per Line](#) (see page 14)
- [ISPF List of REXX Rules](#) (see page 14)
- [REXX Enhancements](#) (see page 15)
- [Add Inactive Option Counterpart to Runtime Options](#) (see page 16)
- [Missing DSN Bypass and Selective RUNTIME Checks](#) (see page 16)
- [Control Card Override](#) (see page 17)
- [Alternate User ID](#) (see page 17)
- [Scenario Documentation](#) (see page 18)

## Installation, Deployment, and Configuration Using CA MSM

You can now install, deploy, and configure CA JCLCheck using CA Mainframe Software Manager™ (CA MSM) Version 4.0. CA MSM already supports installation of CA JCLCheck. Once you deploy an installation of CA JCLCheck, CA MSM helps you complete a basic configuration of the deployed product before you start it.

CA MSM guides you step-by-step through the process of specifying configuration options. For some configurations, CA MSM prompts you to perform certain operations manually, outside of CA MSM.

After you complete the basic configuration, CA MSM creates the CA JCLCheck data sets specific to the configuration you specified. You can further edit the configuration manually.

**Notes:** Documentation changes are listed under "CA JCLCheck Installation" in the Documentation Changes section of the following guide:

- *Installation and Configuration Guide*
- For detailed information, see the *CA Mainframe Software Manager User Guide* and the CA MSM online help.

## Mainframe 2.0 Compliance

The Mainframe 2.0 Compliance enhancement adds new features to meet requirements on how to handle a failing program. Mainframe 2.0 is a comprehensive strategy that unifies the management of CA Technologies mainframe products on z/OS systems.

The following features adhere to Mainframe 2.0 requirements:

- Eye catchers—The eye catcher identifies the ownership of the resources. The eye catcher contains information, such as the latest PTF number that updated the program and the copyright notice. The eye catcher is contained in each CSECT.
- Module Name and Location—CA JCLCheck modules located in CSA are now identified to CA Common Services.
- Product Description Module—A Product Description Module (PDM) is now placed in LPA. The CA-EXAMINE program uses the PDM to analyze and display information about CA JCLCheck, for example, where CA JCLCheck is active.

The following CA Common Services (CCS) release is required to be installed for Mainframe 2.0:

- CCS v14.0 with the following condition:
  - Verify that CAILPA goes into LPA1STxx.

## External Variable Resolution

The external variable resolution enhancement enables CA JCLCheck to validate JCL that contains external variables. This validation is required when a scheduling product uses variables embedded in its JCL. You can configure CA JCLCheck 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 Workload Automation CA 7 Edition, CA Workload Automation ESP Edition, Control-M, and Tivoli Workload Scheduler.



**Note:** Documentation changes are listed under "Resolve External Variables" in the Documentation Changes section of the following guides:

- *Tutorial*
- *Command Reference Guide*
- *Programming Guide*
- *Message Reference Guide*

## DFSORT and ADRDSSU Utility Control Card Support

### Multiple Control Statement Sources

CA JCLCheck now supports control-card checking from multiple control statement sources (ddnames). For example, DFSORT accepts control statements from the following ddnames:

- SYMNames—optional symbol definition statements
- DFSPARM—control statements
- SYSIN—control statements
- JNF1CNTL—optional control statements for JOINKEYS applications
- JNF2CNTL—optional control statements for JOINKEYS applications

The \$UPT macro that the JCLUPT module uses now supports expanded CTL parameter values. Previously, the single specified ddname was required when a value was in the CTL parameter. With CA JCLCheck Version 12.0, the \$UPT macro now supports the following CTL parameter syntax:

```
CTL=(ddname1[{-R|-O|-C}] [, ddname2[{-R|-O|-C}] , . . . )
```

The CTL parameter accepts multiple ddnames. Additionally, a ddname suffix is supported allowing the ddname to be coded as Required, Optional, or Conditional. If no ddname suffix is specified or '-R' is specified, the ddname is required. If '-O' is specified, the ddname is optional. If '-C' is specified, the ddname is conditional. If one or more conditional ddnames are specified, at least one of the group of conditional ddnames must be included in the JCL. For example:

```
$UPT CTL=(SYMNames-O,DFSPARM-C,SYSIN-C,JNF1CNTL-O,JNF2CNTL-O)
```

In this example, the SYMNames, JNF1CNTL, and JNF2CNTL ddnames are optional. If the CTLSCAN option is active, the control statements are listed and processed for each of the specified ddnames. The DFSPARM and SYSIN ddnames are conditional. Either one or both can be specified in the JCL, but at least one of them must be specified in the JCL.

### DFSORT Control Card Syntax Checking

CA JCLCheck now supports syntax checking of all DFSORT control statements when DFSORT is invoked using JCL. Syntax checking of DFSORT control statements is not supported when DFSORT is invoked from another program.

DFSORT performs syntax checking of symbol definitions from SYMNames and all DFSORT statements from DFSPARM and SYSIN. Additionally, syntax checking of DFSORT statements from JNF1CNTL and JNF2CNTL for JOINKEYS applications is supported.

Other Sort products that are compatible with DFSORT and use an alias to the sort program name (for example, SORT) are also supported.

### ADRDSSU Utility Control Card Support

CA JCLCheck now provides support for the ADRDSSU IBM utility. You can now validate the control cards ADRDSSU uses to see if they are correct before the job is run.

**Note:** Documentation changes are listed under "DFSORT and ADRDSSU Utility Control Card Support" in the Documentation Changes section of the following guides:

- *Programming Guide*
- *Command Reference Guide*
- *Installation and Configuration Guide*
- *Message Reference Guide*

## Extended Address Volume Support

Extended Address Volumes (EAV) expand the number of cylinders a volume can have beyond the previous limit of 65,520. CA JCLCheck supports EAV with new parameters that allow the data on these volumes to be retrieved using an expanded work area.

**Note:** Documentation changes are listed under "Extended Address Volume Support" in the Documentation Changes section of the following guide:

- *Message Reference Guide*

## JCLNeat—New REXX Variables

CA JCLCheck now supports new REXX variables for JCLNeat. The JOB.ACCT stem variable is a Job Statement variable for JCLNeat. This enhancement adds two new variables to the JOB.ACCT stem variable: JOB.ACCTNUM and JOB.ACCTINFO. You can use these new variables to specify an account number separately from the accounting information, such as, checking the account code of a job and changing it if necessary.

The DD.DSN stem variable is a DD Statement variable for JCLNeat. This enhancement adds 23 new variables to the DD.DSN stem variable: DD.DSN.0 through DD.DSN.22. You can use these new variables to access individual nodes, such as, checking the second node of a data set name and changing it if necessary.

This enhancement affects only the JCLNeat supported REXX.

**Note:** Documentation changes are listed under "JCLNeat—New REXX Variables" in the Documentation Changes section of the following guides:

- *Programming Guide*
- *Message Reference Guide*

## JCLNeat—Additional REXX Calls and Variables

CA JCLCheck now supports additional calls to REXX for the following new JCL statements:

- NJB
- IF
- PROC
- JCLLIB
- OUTPUT
- DPROC
- SET
- INCLUDE

This enhancement adds over one thousand new variables that you can use with these statements. Additionally, 76 new variables were added for the existing JOB, EXEC, and DD statements.

This enhancement affects only the JCLNeat supported REXX.

**Note:** Documentation changes are listed under "JCLNeat—Additional REXX Calls and Variables" in the Documentation Changes section of the following guide:

- *Programming Guide*

## JCLNeat—Global String Replacement by Statement Type

JCLNeat provides a Global String Replacement (GSR) feature for any character string in the JCL. JCLNeat accepts replacement statements in the GSRCNTL DD statement. These statements let you specify strings of characters to search for in the JCL along with a verb specifying the action to be taken when the string is found. For example, you can use verbs REPL (replace) or DELT (delete).

These statements apply to all the JCL statements in the members being processed. This enhancement allows for a statement type to be specified on the GSR search string to limit the search to only that type of statement. For example, to replace the string 'SYSOUT=A' with 'SYSOUT=\*' you would code the following GSR statements:

```
|SYSOUT=A|REPL|      Find SYSOUT=A and ReplacE with next line.  
|SYSOUT=*|           Replacement value.
```

To limit the change to only DD statements, you can now add a statement type to the end of the search statement:

```
|SYSOUT=A|REPL|DD   Find SYSOUT=A on DD statement.  
|SYSOUT=*|          Replacement with SYSOUT=*
```

This enhancement affects only the JCLNeat global string replacement facility.

**Note:** Documentation changes are listed under "JCLNeat—Global String Replacement by Statement Type" in the Documentation Changes section of the following guide:

- *Command Reference Guide*

## JCLNeat—Options by Statement Type

The JCLNeat feature of CA JCLCheck has many options that affect the formatting of JCL statements. You can now set the following statement formatting options by statement type: Continuation End Column (CEC), Continuation Start Column (CSC), Inline Comment Start Column (ICSC), Operator Start Column (OPSC), and PACK. These options apply to all statements in the JCL. For example, the CSC option specifies the starting column number for the first keyword on each JCL statement line. You can now set different values for different JCL statement types:

- CSCJ=16
- CSCEX=10
- CSCDD=12

**Note:** Documentation changes are listed under "JCLNeat—Options by Statement Type" in the Documentation Changes section of the following guide:

- *Command Reference Guide*
- *Message Reference Guide*

## JCLNeat—REXX Dynamic Option Changes

You can now dynamically change JCLNeat options from a running REXX program with the new JCLNeat feature Dynamic Option Changes. This feature also allows REXX programs to reset JCLNeat options back to their initial run-time settings. When JCLNeat options are changed or reset, the change takes effect immediately on the current statement being processed. The changes remain in effect until changed again, reset, or job termination.

**Note:** Documentation changes are listed under "JCLNeat—REXX Dynamic Option Changes" in the Documentation Changes section of the following guides:

- *Programming Guide*

## JCLNeat—REXX Insert and Delete JCL Statements

You can now easily insert and delete JCL statements from REXX with the new JCLNeat feature Dynamic JCL Statement Insertion and Deletion. This feature provides you with two options that you can use in the REXX program.

The first option is the new REXX variable, `SCA.DELFLG`. This variable deletes JCL statements. When you change the default for `SCA.DELFLG` from N to Y in the REXX program, the current statement is removed from the JCL. This variable is available on every call to the REXX program.

The second option lets you insert statements by using the REXX data stack. JCL statements can be placed on the stack by the REXX program. The statements are later retrieved and added to the JCL being processed. Existing REXX operators are used to place statements on the stack.

This enhancement affects only the JCLNeat supported REXX.

**Note:** Documentation changes are listed under "JCLNeat—Insert and Delete REXX JCL Statements" in the Documentation Changes section of the following guides:

- *Programming Guide*
- *Message Reference Guide*

## JCLNeat PACK Option—Maximum Keywords per Line

CA JCLCheck gives you the option of having each parameter in a JCL statement on a separate line (PACK=N) or packing the parameters on to as few lines as possible (PACK=Y).

This enhancement changes the JCLNeat PACK option to allow you to specify the maximum number of parameters that can be placed on a line. For example, PACK=3 means that no more than three parameters can be packed on to a line of JCL.

For compatibility with earlier releases, PACK=N and PACK=Y are still accepted.

**Note:** Documentation changes are listed under "JCLNeat PACK Option—Maximum Keywords per Line" in the Documentation Changes section of the following guide:

- *Command Reference Guide*

## ISPF List of REXX Rules

CA JCLCheck allows for REXX programs to be used for standards validation in the JCLCheck feature and for JCL customization in the JCLNeat feature. This enhancement adds to those features by allowing for the SYSEXEC library member list to be displayed for easier REXX program selection.

You can now specify an ISPF pattern member name in the existing REXX member fields. The pattern member name can use the asterisk (\*) for multiple character masking and the per cent sign (%) for single character masking. When a pattern character is found in the REXX member name field, the member list for the SYSEXEC library is displayed. Select the member that you want to use from the list and it is filled in as the REXX member name field.

**Note:** Documentation changes are listed under "ISPF List of REXX Rules" in the Documentation Changes section of the following guide:

- *Command Reference Guide*
- *Message Reference Guide*
- *Tutorial*

## REXX Enhancements

CA JCLCheck supports the following three enhancements to JCLNeat REXX:

1. You can now use REXX variable DD.DDSTAT for the CA JCLCheck STDREXX program. DD.DDSTAT is available when the JCLNeat REXXMEM program is called for a DD statement. The DD.DDSTAT variable has flags that indicate information about the DD. These values are:
  - x'08'—Statement is DD DUMMY
  - x'04'—Statement from instream PROC
  - x'02'—Statement is 'DD \*/'DD DATA'
  - x'01'—Statement is unnamed DD
2. You can now use REXX in JCLNEAT without FORMAT=Y. This enhancement allows for FORMAT=B (Basic) to be used to minimize formatting. REXX can now modify the JCL while retaining most of its original formatting. Before this enhancement, specifying the REXXMEM option in JCLNeat forced the format option to be set on.
3. A new variable allows the end of job return code to be set. The \$CA.NEATRC variable lets you set a numeric value. On return from REXX, if the value is not null, the value in the variable is used as a return code for the JCLNeat job step. \$CA.NEATRC is in the list of Miscellaneous Variables.

CA JCLCheck STDREXX now supports the following variables in EDCHEK and they contain information on the member being edited when the JCLCheck command is issued:

- \$CA.IMEMBER
- \$CA.IDSNAME
- \$CA.ITYPE

Before this enhancement, these variables had no value when JCLCheck is called from EDCHEK.

**Note:** Documentation changes are listed under "REXX Enhancements" in the Documentation Changes section of the following guides:

- *Programming Guide*
- *Command Reference Guide*
- *Tutorial*
- *Message Reference Guide*

## Add Inactive Option Counterpart to Runtime Options

This enhancement adds an inactive option counterpart to 54 CA JCLCheck runtime options. You can use the NOxxx option to turn off an option that has been set as a default.

**Note:** Documentation changes are listed under "Add Inactive Option Counterpart to Runtime Options" in the Documentation Changes section of the following guide:

- *Command Reference Guide*
- *Message Reference Guide*

## Missing DSN Bypass and Selective RUNTIME Checks

Missing DSN Bypass—CA JCLCheck now provides support for a new optional parameter in the OPROCLIB and CNTLOVER runtime options. You can now specify the suboption 'C' to indicate CA JCLCheck is to continue validation processing when a library specified in these runtime options is not found. Before this enhancement, if one of the specified libraries did not exist on the system where CA JCLCheck was running, an error occurred and the JCL was not validated. The missing library statement required removal to correct the error and to continue JCL validation. Now when a specified library does not exist, the library is bypassed and JCL validation continues. An information message displays to notify you of the library that was not found.

Selective RUNTIME Checks—CA JCLCheck now provides support for the new BYPRUNT runtime option. The BYPRUNT runtime option has 12 defined suboptions that enable individual runtime checks to be bypassed. Before this enhancement, there was a single option, NORUNTIM, that disabled all runtime checks. This enhancement lets you select runtime checks to bypass.

You can use these two enhancements in invocations of CA JCLCheck from batch, ISPF, and EDCHEK.

**Note:** Documentation changes are listed under "Missing DSN Bypass and Selective RUNTIME Checks" in the Documentation Changes section of the following guides:

- *Command Reference Guide*
- *Message Reference Guide*



## Control Card Override

The control card override enhancement lets you specify a set of override libraries for control card members of system utilities. The override libraries are searched before the library specified in the JCL is validated. You can use this enhancement in invocations of CA JCLCheck from batch, ISPF, and EDCHEK.

**Note:** System utilities refer to utilities that CA JCLCheck validates and syntax checks using the CTLSCAN option, and that are present in JCLUPT.

Typically, control card override is used when CA JCLCheck is called from CA Endeavor (batch or foreground). A control card member can reside in any one of a list of promotion libraries, for example, QA, development, and production. When you use this option, CA JCLCheck searches all promotion levels for the particular control card member. If the member is found in one of these libraries, that member is used for JCL validation. If not found in any of the promotion libraries, the control card member referenced in the actual SYSIN DD (or whichever DD statement the \$UPT CTL keyword points to) is used for JCL validation.

**Note:** Documentation changes are listed under "Control Card Override" in the Documentation Changes section of the following guides:

- *Command Reference Guide*
- *Message Reference Guide*

## Alternate User ID

The Alternate User feature allows for CA JCLCheck validation to be performed under the user ID that the job is run under. CA JCLCheck currently 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.

**Note:** Documentation changes are listed under "Alternate User ID" in the Documentation Changes section of the following guides:

- *Tutorial*
- *Message Reference Guide*

## Scenario Documentation

One of the topics in the *Tutorial* is now written as a use-case scenario. This use case documentation consists of concise, modular information that explains how a specific customer role interacts with a product to achieve a specific goal. Scenarios include a high-level process ("How to...") followed by all the procedures related to that process. Near the beginning of each scenario is a graphic representation of the process that you can follow.

The following scenario appears in the *Tutorial*:

- How to Configure CA JCLCheck to Resolve External Variables

# Chapter 2: Enhancements to Existing Features

---

This section contains the following topics:

[CA MSM](#) (see page 19)

## CA MSM

The Software Configuration Service (SCS) has been added for CA MSM Version 4.0. Use this service to configure CA Technologies mainframe products from the software inventory of the driving system to your target enterprise systems.

The SCS guides you through the configuration creation process, and through the manual steps to implement the configuration. The SCS also includes an address space communications service running on each targeted z/OS system.

**Note:** For more information, see the *CA Mainframe Software Manager User Guide*.



# Appendix A: Known Issues

---

## Published Fixes for Version 12.0.00

All published fixes are available at Published Solutions on CA Support.



# **Appendix B: Third-party Acknowledgements**