Contacting BMC Software

You can access the BMC Software website at http://www.bmc.com. From this website, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

<table>
<thead>
<tr>
<th>Address</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC SOFTWARE INC</td>
<td>713 918 8800</td>
<td>713 918 8000</td>
</tr>
<tr>
<td>2101 CITYWEST BLVD</td>
<td></td>
<td>800 841 2031</td>
</tr>
<tr>
<td>HOUSTON TX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>77042-2827</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Outside United States and Canada

<table>
<thead>
<tr>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>(01) 713 918 8800</td>
<td>(01) 713 918 8000</td>
</tr>
</tbody>
</table>

© Copyright 1999-2015 BMC Software, Inc.

BMC, BMC Software, and the BMC Software logo are the exclusive properties of BMC Software, Inc., are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other BMC trademarks, service marks, and logos may be registered or pending registration in the U.S. or in other countries. All other trademarks or registered trademarks are the property of their respective owners. IT Infrastructure Library® is a registered trademark of the Office of Government Commerce and is used here by BMC Software, Inc., under license from and with the permission of OGC.

ITIL® is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office, and is used here by BMC Software, Inc., under license from and with the permission of OGC.

IBM® Tivoli® Business Service Manager, IBM Tivoli Workload Scheduler, IBM Cognos, IBM InfoSphere DataStage, IBM iSeries, IBM Websphere, and AIX® are the trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

UNIX® is the registered trademark of The Open Group in the US and other countries.

Linux is the registered trademark of Linus Torvalds.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

SAP® R/2 and SAP R/3, SAP Business Objects, and SAP NetWeaver are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

BMC Software considers information included in this documentation to be proprietary and confidential. Your use of this information is subject to the terms and conditions of the applicable End User License Agreement for the product and the proprietary and restricted rights notices included in this documentation.
Restricted rights legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC SOFTWARE INC, 2101 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.

Customer support

You can obtain technical support by using the BMC Software Customer Support website or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, see “Before contacting BMC.”

Support website

You can obtain technical support from BMC 24 hours a day, 7 days a week at http://www.bmc.com/support. From this website, you can:

- Read overviews about support services and programs that BMC offers
- Find the most current information about BMC products
- Search a database for issues similar to yours and possible solutions
- Order or download product documentation
- Download products and maintenance
- Report an issue or ask a question
- Subscribe to receive proactive e-mail alerts when new product notices are released
- Find worldwide BMC support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by telephone or e-mail

In the United States and Canada, if you need technical support and do not have access to the web, call 800 537 1813 or send an e-mail message to customer_support@bmc.com. (In the subject line, enter SupID:<yourSupportContractID>, such as SupID:12345). Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC

Have the following information available so that Customer Support can begin working on your issue immediately:

- Product information
  - Product name
  - Product version (release number)
  - License number and password (trial or permanent)
- Operating system and environment information
  - Machine type
  - Operating system type, version, and service pack or other maintenance level such as PUT or PTF
• System hardware configuration
• Serial numbers
• Related software (database, application, and communication) including type, version, and service pack or maintenance level

 Sequence of events leading to the issue
 Commands and options that you used
 Messages received (and the time and date that you received them)
  • Product error messages
  • Messages from the operating system, such as file system full
  • Messages from related software

License key and password information
If you have questions about your license key or password, contact BMC as follows:
 (USA or Canada) Contact the Order Services Password Team at 800 841 2031, or send an e-mail message to ContractsPasswordAdministration@bmc.com.
 (Europe, the Middle East, and Africa) Fax your questions to EMEA Contracts Administration at +31 20 354 8702, or send an e-mail message to password@bmc.com.
 (Asia-Pacific) Contact your BMC sales representative or your local BMC office.

Third party Software
For the provisions described in the BMC License Agreement and Order related to third party products or technologies included in the BMC Product, see https://docs.bmc.com/docs/display/workloadautomation/Control-M+Workload+Automation+Documentation and click Third-party software (TPS).
# Contents

<table>
<thead>
<tr>
<th>About This Guide</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventions Used in This Guide</td>
<td>14</td>
</tr>
<tr>
<td>Related Publications</td>
<td>17</td>
</tr>
</tbody>
</table>

## Chapter 1 Overview

<table>
<thead>
<tr>
<th>Introduction</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>The verification process</td>
<td>21</td>
</tr>
<tr>
<td>Initialization</td>
<td>21</td>
</tr>
<tr>
<td>Verification</td>
<td>22</td>
</tr>
<tr>
<td>Termination</td>
<td>25</td>
</tr>
<tr>
<td>Installation</td>
<td>25</td>
</tr>
<tr>
<td>JES2/3 Static PROCLIBs</td>
<td>25</td>
</tr>
<tr>
<td>CTJPRC format</td>
<td>26</td>
</tr>
<tr>
<td>Customization</td>
<td>27</td>
</tr>
<tr>
<td>Restrictions</td>
<td>27</td>
</tr>
</tbody>
</table>

## Chapter 2 CTJVER batch utility

<table>
<thead>
<tr>
<th>Parameters</th>
<th>29</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activating the Utility</td>
<td>30</td>
</tr>
<tr>
<td>Batch JCL</td>
<td>33</td>
</tr>
<tr>
<td>Return Codes</td>
<td>33</td>
</tr>
<tr>
<td>CTJVER output report</td>
<td>34</td>
</tr>
<tr>
<td>Examples</td>
<td>34</td>
</tr>
<tr>
<td>Example 1</td>
<td>36</td>
</tr>
<tr>
<td>Example 2</td>
<td>36</td>
</tr>
<tr>
<td>Example 3</td>
<td>37</td>
</tr>
<tr>
<td>Example 4</td>
<td>37</td>
</tr>
<tr>
<td>Example 5</td>
<td>38</td>
</tr>
<tr>
<td>Example 6</td>
<td>39</td>
</tr>
<tr>
<td>Example 7</td>
<td>40</td>
</tr>
<tr>
<td>Example 8</td>
<td>44</td>
</tr>
<tr>
<td>Example 9</td>
<td>45</td>
</tr>
<tr>
<td>Example 10</td>
<td>46</td>
</tr>
</tbody>
</table>

## Chapter 3 Site standards verification

<table>
<thead>
<tr>
<th>Introduction</th>
<th>47</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>49</td>
</tr>
<tr>
<td>Chapter 4</td>
<td>Job flow verification (CTJMRFLOW utility)</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>Parameters</td>
<td>131</td>
</tr>
<tr>
<td>Activating the Utility</td>
<td>133</td>
</tr>
<tr>
<td>Batch JCL</td>
<td>133</td>
</tr>
<tr>
<td>Return Codes</td>
<td>133</td>
</tr>
<tr>
<td>Example</td>
<td>134</td>
</tr>
</tbody>
</table>

Rule selection and searching ........................................... 49
DO block ................................................................. 50
Variables ................................................................. 51
Defining site standard rules online ................................ 53
Loading site standard rules ........................................... 60
Verifying site standards using REXX EXECs .......................... 61
Rule Definition Facility ................................................ 63
Entry Panel ............................................................... 66
Table List Screen ....................................................... 69
Rule List Screen ......................................................... 71
Copying Rules to Another Table ...................................... 74
Rule Definition Screen ................................................ 76
Commands of the Rule Definition Screen ............................. 82
Editing Rule Definitions in the Edit Environment .................. 82
Exiting the Rule List Screen ......................................... 84
Copying Rules to Another Table ...................................... 85
Parameter Descriptions .................................................. 87
CONTINUE SEARCH: General Parameter .................................. 88
DESC: General Parameter ................................................ 89
DO statement: Action Parameter ....................................... 90
DO IF / DO ELSE / DO ENDIF: Action Parameter ...................... 92
DO MSG: Action Parameter .............................................. 95
DO RETURN: Action Parameter ........................................... 96
DO REXX: Action Parameter ............................................ 97
DO SET: Action Parameter ............................................. 98
ENVIRONMENT: General Parameter .................................... 107
JOBNAME: Common Selection Parameter ................................ 109
LIBRARY: Common Selection Parameter ................................ 110
MEMBER: Common Selection Parameter ................................ 111
ON DD: Event Selection Parameter .................................. 112
ON DDDSIN: Event Selection Parameter ............................... 115
ON DDSYSOUT: Event Selection Parameter ............................ 116
ON EXEC: Event Selection Parameter ................................ 117
ON INCLUDE: Event Selection Parameter .............................. 119
ON JCLLIB: Event Selection Parameter ................................. 120
ON JOB: Event Selection Parameter .................................. 121
ON JOBEND: Event Selection Parameter .............................. 123
ON JOBINIT: Event Selection Parameter ............................... 124
ON PROC: Event Selection Parameter ................................ 125
RULE: General Parameter ............................................... 126
SCHDLIB: Common Selection Parameter ................................ 127
SCHDTAB: Common Selection Parameter ................................ 128
SCHDLIB: Common Selection Parameter ................................. 127
SCHDTAB: Common Selection Parameter ................................ 128
Appendix A Messages

Appendix B Sample exit 4

Appendix C Editing Rule Definitions in the IOA Edit Environment

Line Editing Commands ........................................... 236
Maintaining Valid Rule Definitions.......................... 238

Appendix D The %%$PARSE function

Overview of %%%PARSE ........................................... 247
Parsing Words .......................................................... 248
Using Dummy Variables (Place Holders) ..................... 249
Using Patterns in Parsing ......................................... 250
  Using String Patterns ......................................... 250
  Using Numeric Patterns Within the Template .......... 252
  Using More Than One Pattern and Combining Pattern Types in the Template 254

Index ................................................................. 259
Tables

CTJVER PROCEDURE parameters .................................................. 30
CTJVER SYSIN statement keywords .............................................. 32
CTJVER Return Codes ................................................................. 34
System variables ......................................................................... 51
Date system variables .................................................................. 52
General Parameters ..................................................................... 56
Common Selection Parameters ...................................................... 56
Event Selection Parameters (ON Parameters) ............................... 57
Action Parameters ....................................................................... 58
REXX EXEC call functions ............................................................ 61
Control-M/JCL Verify Rule Definition Facility Screens .................. 63
Options of the Table List Screen .................................................. 70
Rule Definition Facility Rule Types ............................................... 72
Commands of the Rule List Screen ............................................... 73
Options of the Rule List Screen .................................................... 74
Fields of the Rule List Screen Copy Window .................................. 75
General Parameters ..................................................................... 77
Common Selection Parameters ...................................................... 78
ON statement options ................................................................... 80
ON Statement Or/Not Subparameter ............................................. 80
Commands of the Rule Definition Screen ...................................... 82
Commands for Exiting the Rule Definition Screen ......................... 83
Fields in the Window for Copying Rules to Another Table ............... 86
CONTINUE SEARCH Values ............................................................ 88
DO Actions .................................................................................... 90
Logical Operators for IF Statement ................................................. 92
Boolean Operators for IF Statement ............................................... 93
Built-in functions ......................................................................... 98
ON DD Subparameters ................................................................ 112
ON DD variables ......................................................................... 113
ON EXEC Subparameters ............................................................. 117
ON EXEC variables ..................................................................... 117
ON EXEC user-defined variables .................................................. 117
ON JOB Subparameters ............................................................... 121
ON JOB variables ...................................................................... 121
CTJMRFLOW PROCEDURE parameters ........................................ 131
CTJMRFLOW JOBLIST.DAJOB statement keywords .................... 132
CTJPLAN PROCEDURE parameters .............................................. 136
CTJXVER Edit Macro Return Codes .............................................. 147
Control-M/JCL Verify verification criteria ...................................... 157
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTJVER options for MEMBER field</td>
<td>159</td>
</tr>
<tr>
<td>Options and commands for verification from Control-M interfaces</td>
<td>161</td>
</tr>
<tr>
<td>Return Codes</td>
<td>162</td>
</tr>
<tr>
<td>Macro keywords</td>
<td>167</td>
</tr>
<tr>
<td>Internal messages</td>
<td>171</td>
</tr>
<tr>
<td>CTJINIT functions</td>
<td>175</td>
</tr>
<tr>
<td>CTJINIT Return Codes</td>
<td>178</td>
</tr>
<tr>
<td>Delete Commands</td>
<td>236</td>
</tr>
<tr>
<td>Copy commands</td>
<td>237</td>
</tr>
<tr>
<td>Move commands</td>
<td>237</td>
</tr>
<tr>
<td>Repeat commands</td>
<td>237</td>
</tr>
<tr>
<td>Insert commands</td>
<td>237</td>
</tr>
<tr>
<td>Location commands</td>
<td>238</td>
</tr>
</tbody>
</table>
Figures

Sample CTJPRC member for JES2 ................................................................. 26
Sample CTJPRC member for JES3 ................................................................. 27
CTJVER – Example 1 .................................................................................. 36
CTJVER – Example 2 .................................................................................. 36
CTJVER – Example 3 .................................................................................. 37
Verified job - Sample00 – Example 4 .......................................................... 37
CTJVER utility – Example 4 ......................................................................... 37
CTJVER utility - SYSPRINT – Example 4 ..................................................... 38
Verified job - Sample01 – Example 5 ........................................................... 39
CTJVER utility – Example 5 ......................................................................... 39
CTJVER utility - SYSPRINT – Example 5 ..................................................... 39
SMART Table with variable %EX_SYSTEM – Example 6 ................................. 40
Job definition with variable %EXDATE – Example 6 ..................................... 40
Job with three statements to be resolved – Example 6 ................................. 41
Verified job - Sample08 – Example 6 .......................................................... 41
CTJVER utility - SYSPRINT – Example 6 ..................................................... 42
Verified job - Sample01 – Example 7 ........................................................... 43
CTJVER utility - SYSPRINT – Example 7 ..................................................... 43
IEBGENER – Example 8 ............................................................................. 45
SORT – Example 9 ..................................................................................... 45
IEBCOPY – Example 10 ............................................................................. 46
Rule Definition Screen ............................................................................. 54
Example of REXX call ............................................................................. 62
Rule Definition Facility Entry Panel ........................................................... 66
Rule Definition Facility Entry Panel Search Window.................................... 68
Rule Definition Facility Table List Screen .................................................. 69
Rule Definition Facility Delete Table Confirmation Window ......................... 71
Rule Definition Rule List Screen ................................................................ 72
Rule List Screen Copy Window .................................................................. 75
Rule Definition Screen ............................................................................. 77
General Parameters ................................................................................. 77
Common Selection Parameters .................................................................. 78
Event Selection Parameters ...................................................................... 79
Entering Editing Commands ...................................................................... 83
Rule List Screen Exit Option Window ........................................................... 84
Window for Copying Rules to Another Table ................................................. 85
IF/ ELSE/ ENDF Statements Format ............................................................ 92
DO IF example 1 ....................................................................................... 94
DO MSG Parameter Example .................................................................... 95
DO RETURN example ............................................................................... 96
About This Guide

This guide contains the information necessary to help you verify JCL jobs. The guide contains the following parts:

Chapter 1 – Overview

Provides an introduction to Control-M/JCL Verify.

Chapter 2 – CTJVER batch utility

Provides instructions for using the CTJVER utility to perform batch verifications of JCL jobs and Control-M job definitions.

Chapter 3 – Site standards verification

Provides instructions for using Control-M/JCL Verify to ensure that JCL statements comply with required enterprise standards.

Chapter 4 – Job flow verification

Provides instructions for using Control-M/JCL Verify to validate work flows through the verification of job order and dependencies.

Chapter 5 – CTJPLAN utility

Provides instructions for using the CTJPLAN utility to verify the JCL statements contained in Control-M job definitions that are about to be submitted.

Chapter 6 – CTJVER edit macro

Provides instructions for using the CTJVER edit macro to verify JCL jobs and Control-M job definitions.

Chapter 7 – Online job verification from IOA and Control-M interfaces

Provides instructions for invoking Control-M/JCL Verify from IOA and Control-M user interfaces for verifying JCL jobs and Control-M job definitions.
Chapter 8– The Control-M/JCL Verify Application Program Interface (API)

Provides instructions for using the API so that JCL verifications can be performed from user programs.

Chapter 9 – CTJINIT utility

Provides instructions for using the CTJINIT utility to boost the performance of JCL verification and to load the site standards rules.

Appendix A – Messages

Provides descriptions of three levels of messages - Information, Warning, and Error.

Appendix B – Sample exit 4

Provides information about a sample, which is included with Control-M/JCL Verify, that uses CTJAPI.

Appendix C – Editing Rule Definitions in the IOA Edit Environment

Provides instructions for using IOA line editing commands for editing rule definitions within the IOA Edit environment.

Conventions Used in This Guide

Notational conventions that may be used in this guide are explained below.

Standard Keyboard Keys

Keys that appear on the standard keyboard are identified in boldface, for example, Enter, Shift, Ctrl+S (a key combination), or Ctrl S (a key sequence).
**WARNING**

The commands, instructions, procedures, and syntax illustrated in this guide presume that the keyboards at your site are mapped in accordance with the EBCDIC character set. Certain special characters are referred to in this documentation, and you must ensure that your keyboard enables you to generate accurate EBCDIC hex codes. This is particularly true on keyboards that have been adapted to show local or national symbols. You should verify that

$s$ is mapped to $\text{x'5B'}$

$\#$ is mapped to $\text{x'7B'}$

@ is mapped to $\text{x'7C'}$

If you have any questions about whether your keyboard is properly mapped, contact your system administrator.

---

**Preconfigured PFKeys**

Many commands are preconfigured to specific keys or key combinations. This is particularly true with regard to numbered PF keys, or pairs of numbered PFKeys. For example, the END command is preconfigured to, and indicated as, PF03/PF15. To execute the END command, press either the PF03 key or the PF15 key.

Instructions to enter commands may include

- only the name of the command, such as, enter the END command
- only the PF keys, such as, press PF03/PF15
- or both, such as, press PF03/PF15, or enter the END command

**Command Lines and Option Fields**

Most screens contain a command line, which is primarily used to identify a single field where commands, or options, or both, are to be entered. These fields are usually designated COMMAND, but they are occasionally identified as COMMAND/OPT or COMMAND/OPTION.

Option field headings appear in many screens. These headings sometimes appear in the screen examples as OPTION, or OPT, or O.

**Names of Commands, Fields, Files, Functions, Jobs, Libraries, Members, Missions, Options, Parameters, Reports, Subparameters, and Users**

The names of commands, fields, functions, jobs, libraries, members, missions, options, parameters, reports, subparameters, users, and most files, are shown in standard UPPERCASE font.
Conventions Used in This Guide

User Entries

In situations where you are instructed to enter characters using the keyboard, the specific characters to be entered are shown in this **UPPERCASE BOLD** text, for example, type **EXITNAME**.

Syntax statements

In syntax, the following additional conventions apply:

- A vertical bar ( | ) separating items indicates that you must choose one item. In the following example, you would choose a, b, or c:
  
  a | b | c

- An ellipsis ( . . . ) indicates that you can repeat the preceding item or items as many times as necessary.

- Square brackets ([ ]) around an item indicate that the item is optional. If square brackets ([ ]) are around a group of items, this indicates that the item is optional, and you may choose to implement any single item in the group. Square brackets can open ([) and close (]) on the same line of text, or may begin on one line of text and end, with the choices being stacked, one or more lines later.

- Braces ({ }) around a group of items indicates that the item is mandatory, and you must choose to implement a single item in the group. Braces can open ({) and close (}) on the same line of text, or may begin on one line of text and end, with the choices being stacked, one or more lines later.

Screen Characters

All syntax, operating system terms, and literal examples are presented in this typeface. This includes JCL calls, code examples, control statements, and system messages. Examples of this are:

- calls, such as
  
  CALL 'CBLTDLI'

- code examples, such as
  
  FOR TABLE owner.name USE option, . . . ;

- control statements, such as
system messages, both stand-alone, such as You are not logged on to database database_name, and those embedded in text, such as the message You are not logged on to database database_name, are displayed on the screen.

Variables

Variables are identified with italic text. Examples of this are:

- In syntax or message text, such as
  Specify database database_name
- In regular text, such as
  replace database database_name1 with database database_name2 for the current session
- In a version number, such as
  EXTENDED BUFFER MANAGER for IMS 4.1.xx

Special elements

This book includes special elements called notes and warnings:

--- NOTE ---
Notes provide additional information about the current subject.

--- WARNING ---
Warnings alert you to situations that can cause problems, such as loss of data, if you do not follow instructions carefully.

Related Publications

INCONTROL for z/OS Administrator Guide

Information for system administrators about customizing and maintaining INCONTROL™ products.
Related Publications

**INCONTROL for z/OS Installation Guide: Installing**

Step-by-step guide to installing INCONTROL products using the INCONTROL Installation and Customization Engine (ICE) application.

**INCONTROL for z/OS Installation Guide: Customizing**

Step-by-step guide for customizing INCONTROL products using the INCONTROL Installation and Customization Engine (ICE) application.

**INCONTROL for z/OS Messages Manual**

Comprehensive listing and explanation of all INCONTROL and IOA messages and codes.

**INCONTROL for z/OS Security Guide**

Step-by-step guide to implementing security in INCONTROL products.

**INCONTROL for z/OS Utilities Guide**

Describes utilities designed to perform specific administrative tasks that are available to INCONTROL products.

**Control-M for z/OS User Guide**

Guide for using Control-M for z/OS.
Overview

This chapter includes the following topics:

Introduction ................................................................. 19
The verification process .................................................. 21
  Initialization ................................................................. 21
  Verification ................................................................. 22
  Termination ................................................................. 25
Installation ...................................................................... 25
  JES2/3 Static PROCLIBs ................................................. 25
  CTJPRC format ............................................................ 26
Customization .................................................................. 27
Restrictions ..................................................................... 27

Introduction

Control-M/JCL Verify is a new product in the IOA family of z/OS products. Control-M/JCL Verify validates JCL jobs, as described later in this section, and issues validation reports. While Control-M/JCL Verify is closely integrated with Control-M for z/OS, it can be installed as a standalone product on sites that are not using Control-M.

Control-M/JCL Verify is capable of validating:

- JCL jobs

  Members that include JCL statements are analyzed to ensure that the JCL statements are valid.
Control-M job definitions

— Members, which Control-M job definitions refer to by MEMNAME and MEMLIB, are analyzed to ensure that the JCL statements are valid.

— In-stream JCL statements, which are included in the job definition itself, are analyzed to ensure that they are valid.

Control-M/JCL Verify verifies the following issues:

■ basic JCL syntax

■ JES2 and JES3 statements

■ syntax of the IEBGENER, IEBCOPY, and SORT utilities

■ dataset existence

■ job flow and dependencies

■ dataset access privileges

■ load module existence

Control-M/JCL Verify consists of the following features:

■ a batch utility (CTJVER) for verifying JCL jobs and Control-M job definitions

■ a batch utility (CTJPLAN) for verifying Control-M jobs that are scheduled for submission

■ an edit macro, which can be invoked from ISPF, for verifying a job being edited

■ new Control-M/JCL Verify JV option available from the IOA Primary Option Menu

■ new line options and commands for invoking Control-M/JCL Verify from Control-M panels

■ a Control-M/JCL Verify application program interface (API) for performing JCL verification from user applications

■ a batch utility (CTJMRFLW) for dependency awareness verification
The verification process

Chapter 1 Overview

The verification process

The verification process consists of the following steps:

- Initialization
- Verification
- Termination

Initialization

In the initialization step, the IOA and basic environment for Control-M/JCL Verify is created.

This step includes:

- setting the Control-M/JCL Verify functionality by specifying
  - the installation parameters
  - the parameters supplied during runtime
- initializing the security
- collecting resources that are required for JCL verification

Since the required resources are common for all Control-M/JCL Verify users, JCL verification performance is usually significantly increased by

1. creating the Control-M/JCL Verify Shared Object and loading the site standards rules to the common area (see Chapter 9, “CTJINIT utility”)

2. identifying JES2 PROCLIBs by
A. determining if dynamic PROCLIBs are in use

B. determining if mandatory PROCLIBs are in use

C. checking CTJPRC for static PROCLIBs. For more details on the CTJPRC member, see “Installation” on page 25.

If the mandatory JES2 PROCLIBs DD statements do not exist, the Control-M/JCL Verify initialization will fail, and the verification will be stopped.

3. loading site related information into the shared objects

Verifications

The verification consists of the following stages:

- JCL and syntax verification
- Environment verification
- Site standards verification

JCL and syntax verification

In this stage, the validity of the following items is verified:

- the JCL used in the job
- the syntax of the JES2 or JES3 statements
- the syntax of the IEBGENER, IEBCOPY, and SORT utilities

The verification process continues to the next stage if there are no JCL errors and no JES2 or JES3 syntax errors that would cause JES2 or JES3 to reject the job.

Environment verification

In this stage, the validity of the following items is verified in the environment where the job is to be executed:

- File Access
- File characteristics
- Program existence
- Dependencies and job flow verification
File Access

The File Access Validation determines whether the user has access to the dataset required by the job being verified.

The user being checked, who might not necessarily be the one performing the verification, is determined by the following criteria:

- the user that is specified by the USER keyword in the JOB statement
- in Control-M, the user that is specified by the OWNER field in the job definition
- in the CTJVER utility, the user that is specified by the USER statement
- in ISPF, the user that is specified in the CTJXVER P edit macro window (see “Verification mode” on page 146)
- the user that performs the verification

The authority validations checks are base on the DISP keyword in the DD statement as listed below:

- DISP=SHR – required read permission
- DISP=OLD – required update permission
- DISP=NEW – required create permission
- DISP=MOD and the file exists – required update permission
- DISP=MOD and the file is new – required create permission
- DISP=(…,DELETE) – required delete permission
- DISP=(…,UNCATLG) – required uncatalog permission

The READ, UPDATE, CREATE, DELETE and UNCATLG permissions are logical and depend on the security package that is installed and on the z/OS release.

File characteristics

Based on DISP, UNIT, and VOL DD statement keywords, verifies that the file

- exists
- is cataloged
- exists in the correct volume
Program existence

The name of the module in the EXEC PGM= statement is verified that it exists in the STEPLIB, JOBLIB, or LINKLIST libraries.

NOTE

Control-M/JCL Verify checks program existence using IBM search order rules.

Dependencies and job flow verification

In addition to single job verification, Control-M/JCL Verify can validate work flows through the verification of the exact job order and dependencies.

NOTE

Control-M/JCL Verify can verify a mixture of Control-M schedule definition and regular JCL jobs.

A procedure, named CTJMRFLW, invokes Control-M to determine the exact job order and job dependencies. Then, after arranging the jobs and schedule definitions in the exact order as they are intended to run, CTJMRFLW invokes Control-M/JCL Verify for the job flow verification. At each stage of the job flow, Control-M/JCL verify examines the datasets of the previously verified job to determine if those datasets still exist or were deleted, and, based on the dependencies, verifies the job flow.

Site standards verification

Site standards rules, which were previously defined and created through the Control-M/JCL Verify rules definition (JR screen), are loaded through CTJINIT.

The site standards verification is performed for each JCL statement.
Termination

1. Sets the following return codes based on the verification results:
   - Return code 4
     The results contain only failures (for example, file not found) that are caused by the current environment and can be set correctly before the job is executed.
   - Return code 8
     The results contain failures (for example, a JCL error) that require changes in the JCL.
   - Return code 12
     The results contain failures that are caused by the Control-M/JCL Verify input, which stops the verification process.
   - Return code 16 and above
     The results contain failures that are caused by Control-M/JCL Verify processing errors (for example, REGION is too small).

2. Print the verification results or prepare the results in a buffer, which can be accessed by a user application with the Control-M/JCL Verify application program interface (CTJAPI).

3. If there are no more verification requests, perform a cleanup and then release all resources.

Installation

Follow the installation as described in the INCONTROL for z/OS Installation Guide: Installing.

JES2/3 Static PROCLIBs

JES2/3 PROCLIBs can be defined in two ways:
1. PROCxx DD statements in the JES2 startup procedure, or IATPLBxx DD statements in the JES3 startup procedure. The PROC00 in JES2 and IATPLBST in JES3 are mandatory statement in JES2 and JES3 respectively.

2. Dynamic definition in the JES PARM member or by the JES2 command $ADD PROCLIB, or *MODIFY in JES3. When dynamic PROCLIBs are used, they override the static PROCLIBs.

Control-M/JCL Verify can identify automatically the dynamic PROCLIBs, but cannot identify the static PROCLIBs.

In order to determine the list of static PROCLIBs, Control-M/JCL Verify reads member CTJPRC in the IOA PARM library.

CTJPRC contains the list of DD statements with JES2/3 static PROCLIBs.

CTJPRC format

1. JES2/3 PROCLIB DD name in position 1

2. Procedure library name follows after at least one blank. The library name must be a valid DSNAME.

3. "*" (asterisk) in position 1 is a comment line.

The following system symbol can be embedded within the library name:

- &SYSNAME
- &USERID
- &SYSPLEX
- &SMFID

Note that a dot (\'\.') must be followed after the system name if the library name has a suffix right after the system name. For example: SYS2.&SYSNAME..PROCLIB

Following are the sample CTJPRC members that are in the IOA PARM library:

**Figure 1 Sample CTJPRC member for JES2**

```plaintext
**===================================================================**
**                                                                   **
**  This member contains the JES2 procedure libraries referenced     **
**  with the associated DD names. The procedure libraries should be  **
**  specified in the same order and with the same DD name as they    **
**  appear in the JES2 JCL procedure.                               **
**  The rules to insert a line are:                                 **
**  o DD name in position 1                                         **
**  o Procedure library name follows after at least one blank       **
**===================================================================**
```
Follow the instructions for customizations described in the INCONTROL for z/OS Installation Guide: Customizing.

Restrictions

Control-M/JCL Verify only verifies JCL in partitioned datasets (PDS or PDSE). The partitioned dataset must have a record length of 80, and a fixed record format (FB/F).
CTJVER batch utility

The CTJVER utility is capable of verifying a JCL job before it is submitted. The utility can also verify jobs referred to from or contained inside Control-M definitions. CTJVER can either verify a single job or work in mass mode, verifying many jobs together (for example, verifying all the jobs in a library).

This chapter includes the following topics:

Parameters ............................................................ 30
Activating the Utility .................................................. 33
Batch JCL .................................................................. 33
Return Codes ............................................................ 34
CTJVER output report ................................................ 34
Examples ................................................................. 36
Example 1 ............................................................... 36
Example 2 ............................................................... 36
Example 3 ............................................................... 37
Example 4 ............................................................... 37
Example 5 ............................................................... 38
Example 6 ............................................................... 39
Example 7 ............................................................... 43
Example 8 ............................................................... 44
Example 9 ............................................................... 45
Example 10 ............................................................. 46
Parameters

Table 1 describes the parameters that the utility receives from the JCL procedure:

**Table 1**  
**CTJVER PROCEDURE parameters (part 1 of 2)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| JES       | Whether to verify that the JES2 or JES3 statements are correct. Valid values are:  
- Y - Yes  
- N - No  
- D - as defined by the JESTTMNT parameter in the CTJPARM member. Default  |
| FE        | Whether to verify file existence. Valid values are:  
- Y - Yes  
- N - No  
- D - as defined by the DSNEXIST parameter in the CTJPARM member. Default  |
| FA        | Whether to verify file access privileges. Valid values are:  
- Y - Yes  
- N - No  
- D - as defined by the DSNACCSS parameter in the CTJPARM member. Default  |
| STDR      | Whether to use the Control-M/JCL Verify rules to verify site standards. Valid values:  
- Y - Yes  
- N - No  
- D - as defined by the STDR parameter in the CTJPARM member. Default  |
| ME        | Whether to verify that the load modules exist. Valid values are:  
- Y - Yes  
- N - No  
- D - as defined by the PGMCHECK parameter in the CTJPARM member. Default  |
Table 1  CTJVER PROCEDURE parameters (part 2 of 2)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| ML | The minimum level of message severity to be issued. Valid values are:  
  - I – Information – all messages are issued.  
  - W – warning messages and errors are issued.  
  - E – Only error messages are issued  
  - D - as defined by the MSGLEVEL parameter in the CTJPARM member. Default |
| CTM | Whether to resolve Control-M AutoEdit variables in the job. Valid values are:  
  - Y - Yes  
  - N - No  
  - D - as defined by the CTMVAR parameter in the CTJPARM member. Default |
| UT | Whether to verify the syntax of the IEBGENER, IEBCOPY, and SORT IBM utilities. Valid values are:  
  - Y - Yes  
  - N - No  
  - D - as defined by the SUPUTIL parameter in the CTJPARM member. Default |

Note: UT=Y must be used with ME=Y, FE=Y, and FA=Y.

If any of the PROCEDURE parameters are omitted, the default values are used.

Table 2 describes the keywords used in the SYSIN statements to perform various verification tasks. The table indicates which keywords are used for each task.
### Table 2  CTJVER SYSIN statement keywords (part 1 of 2)

<table>
<thead>
<tr>
<th>To perform this task:</th>
<th>Use these keywords:</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>To specify which user is used in the verifications:</strong></td>
<td>USER</td>
<td>The user ID to be used in the file access privileges (FA) verifications. Valid values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- user_ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DEFAULT - use the default user ID (either the user invoking the CTJVER utility, or in the case of a Control-M job definition, the Control-M owner)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default user ID is determined according the criteria described in “File Access” on page 23.</td>
</tr>
<tr>
<td><strong>To specify the odate used for odate variable resolutions:</strong></td>
<td>ODATE</td>
<td>The odate to be used in odate variable resolutions. The valid format is: YYMMDD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: The ODATE value is used in the SCHEDLIB statements that follow it, until a new ODATE is specified.</td>
</tr>
<tr>
<td><strong>To specify the environment rules used in the verifications:</strong></td>
<td>ENV</td>
<td>The rules that are defined for the specified environment, in addition to the GENERAL rules, are to be used in site standard verifications. The ENV keyword applies to all the verifications until the ENV is redefined as shown in the following example:</td>
</tr>
</tbody>
</table>
| | | //SYSIN DD *  
| | | USER=M01  
| | | ODATE=ODAT  
| | | ENV=PROD  
| | | LIB=MY.PROD.LIB,MEM=*  
| | | ENV=TEST  
| | | LIB=MY.TEST.LIB,MEM=*  
| | | /*  
| | | Note: The default is blank, meaning that only the GENERAL rules are used for the site standard verifications. |
| **To verify jobs in their library:** | LIB | JCL library name where the JCL job to be verified is located. Mandatory. |
| | MEM | Member name where the JCL job to be verified is located. Character masking is supported so that “MEM=∗” indicates that all the jobs in the library are to be verified. |
Activating the Utility

You can activate the utility through a batch JCL.

Batch JCL

The following is a sample batch JCL used to invoke CTJVER:

Table 2  CTJVER SYSIN statement keywords (part 2 of 2)

<table>
<thead>
<tr>
<th>To perform this task:</th>
<th>Use these keywords:</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To verify jobs referred to from or contained inside Control-M definitions:</td>
<td>SCHEDLIB</td>
<td>Name of the library containing the job scheduling definition to be verified. Mandatory.</td>
</tr>
<tr>
<td></td>
<td>TABLE</td>
<td>Name of the table containing the job scheduling definition to be verified. Character masking is supported so that “TABLE=<em>” indicates that all the jobs in all the tables in the scheduling library are to be verified. In this case, “JOB=</em>” can be omitted since “JOB=*” is assumed.</td>
</tr>
<tr>
<td></td>
<td>JOB</td>
<td>Name of the job scheduling definition to be verified. Character masking is supported so that “JOB=*” indicates that all the jobs in the table are to be verified.</td>
</tr>
<tr>
<td>To verify ordered jobs:</td>
<td>ORDERID</td>
<td>Specifies the order ID of the job in the AJF that will be verified.</td>
</tr>
</tbody>
</table>

The SCHEDLIB, TABLE, JOB, and ORDERID keywords are available only when Control-M is installed.

**NOTE**

- If there is not enough room in the input record for the entire CTJVER statement, enter any non-blank character in position 72, to indicate that the following record is a continuation of the current record, and continue the statement in the following record.

- Regarding the SCHEDLIB statement: If MULTJOBS=Y is specified in CTMPARM, only the first job will be submitted from the JCL member by Control-M. Therefore, if the JCL member to be verified contains more than one job, only the first job will be verified.
Return Codes

Table 3  CTJVER Return Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Operation performed successfully.</td>
</tr>
<tr>
<td>4</td>
<td>Control-M/JCL Verify discovered that the verified resource is not in the required status at the time of the verification. For example, an input dataset does not exist. Review the status and decide if there is a need to make a change in the job.</td>
</tr>
<tr>
<td>8</td>
<td>Control-M/JCL Verify discovered a problem in the job that will cause it to fail with a JCL error or to be rejected by JES2 or JES3, either during job submission or execution. Correct the JCL in the job.</td>
</tr>
<tr>
<td>12</td>
<td>Error in the CTJVER parameters or statements. Review the previous error messages in JOBLOG and SYSPRINT, and then correct the parameters or the control statements.</td>
</tr>
<tr>
<td>14</td>
<td>Member is already in use.</td>
</tr>
<tr>
<td>16 and above</td>
<td>Internal error in the Control-M/JCL Verify product. Try again. If the problem reoccurs, contact BMC Customer Support.</td>
</tr>
</tbody>
</table>

CTJVER output report

The report includes

- the JCL listing as received from z/OS
- the messages issued by z/OS
- lines with AutoEdit variables before and after they are resolved, each on a separate line
messages issued by JVER which are separated into 3 severity types: I-Information, W-Warning, and E-Error.

The user can choose the minimum level of messages to be issued using the CTJPARM parameter MSGLEVEL, which can be overridden from the interfaces. These messages are issued only after the job has passed the valid syntax check.

For example of output reports see “Example 4” on page 37 and “Example 5” on page 38.
Examples

Example 1

In the following example, CTJVER verifies all the jobs in the JOB1 member located in the XX.YY library.

JES2 or JES3 statements are verified or not verified depending on JESTTMNT in CTJPARM. The existence of the dataset will be verified. The M01 user is verified for file access privileges. The existence of the load modules will not be verified. The minimum level of message severity to be issued depends on how MSGLEVEL is defined in CTJPARM.

Figure 3  CTJVER – Example 1

```plaintext
// . . . JOB . . .
//      EXEC CTJVER,JES=D,FE=Y,FA=Y,ME=N
//SYSIN DD *
USER=M01
LIB=XX.YY,MEM=JOB1
/*
//
```

Example 2

In the following example, CTJVER verifies that all the jobs in all the members located in the XX.YY.ZZ library. The M01 user is verified for file access privileges. The existence of load modules will not be verified.

Figure 4  CTJVER – Example 2

```plaintext
// . . . JOB . . .
//      EXEC CTJVER,JES=D,FE=Y,FA=Y,ME=N
//SYSIN DD *
USER=M01
LIB=XX.YY.ZZ,MEM=* 
/*
//
```
Example 3

In the following example, CTJVER verifies the following:

- the ODATE of all jobs is set to December 1, 2013
- all jobs from table TAB1 and library AA.CC
- all jobs from all tables in library AA.DD

The JOB’s definition owner is the user ID for verifying the file access privileges.

```
// . . . JOB . . .
//      EXEC CTJVER,JES=D,FE=Y,FA=Y,ME=N
//SYSIN DD *
ODATE=131201
USER=*DEFAULT
SCHEDLIB=AA.BB,TABLE=TAB1,JOB=JOB1
SCHEDLIB=AA.CC,TABLE=TAB1,JOB=*
SCHEDLIB=AA.DD,TABLE=*
/*
```

Example 4

The following example shows a CTJVER output report for a job with Control-M/JCL Verify messages.

Figure 6 shows the job that is to be verified.

```
//IEFBR14 JOB,BR14,MSGCLASS=X,NOTIFY=N18A,MSGLEVEL=(1,1)
//S1 EXEC PGM=IEFBR14,COND=EVEN
//NEWTEMP DD SPACE=(TRK,1),DSN=NEW.FILE.NO.DISP
//FILEOK DD DISP=SHR,DSN=CTJP.V800.JCL.SAMPLES
```

Figure 7 shows the job that performs the verification.

```
//SAMPLE00 JOB,YY,CLASS=A,MSGCLASS=X,
    NOTIFY=&SYSUID
// JCLLIB ORDER=IOAP.V800.PROCLIB
// INCLUDE MEMBER=IOASET
//SAMPLE1 EXEC CTJVER
//SYSIN DD *
    LIB=CTJP.V800.JCL.SAMPLES MEM=IEFBR14
/*
```
The CTJVER utility output report, shown in Figure 8, includes various messages. To simplify the reading and interpretation of the messages, Control-M/JCL Verify combines the messages that are issued by the system in the JESYSMSG member with the JCL statements.

The CTJVER utility output report includes CTJVER batch utility messages, which are indicated by a CTJU prefix. The first CTJU01I message displays the CTJVER PROCEDURE parameters, which in this example are specified as defined in CTJPARM. The other CTJU01I messages are input statement echo messages.

The normal JCL statements are displayed similar to how they appear in the JOB JESJCL output.

The user set the ME (Module Existence) option to the default value by setting the PGMCHECK parameter to Y in the CTJPARM member. As a result, the CTJP0GI message is displayed. The message indicates that the IEFBR14 load module exists in SYS1.LINKLIB, one of the LINKLIST libraries.

The CTJ009I message is issued for each verified job. For members with multiple jobs, the message is displayed after each job.

The CTJ003I message is issued for each control statement.

If no warning or error messages are issued, the validation ends with return code (RC) zero.

Figure 8  CTJVER utility - SYSPRINT – Example 4

<table>
<thead>
<tr>
<th>CTJU01I</th>
<th>FA=D FE=D JES=D ME=D ML=D</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTJU01I</td>
<td>LIB=CTJP.V800.JCL.SAMPLES MEM=IEFBR14</td>
</tr>
<tr>
<td>CTJ002I</td>
<td>Start verifying member IEFBR14 DSNAME CTJP.V800.JCL.SAMPLES</td>
</tr>
<tr>
<td>CTJ008I</td>
<td>Start verifying JOB IEFBR14 MEMBER IEFBR14</td>
</tr>
<tr>
<td></td>
<td>1 //IEFBR14 JOB .BRI14.MSGCLASS=X,NOTIFY=N18A,MSGLEVEL=(1,1)</td>
</tr>
<tr>
<td></td>
<td>2 //S1 EXEC PGM=IEFBR14,COND=EVEN</td>
</tr>
<tr>
<td>CTJP0GI</td>
<td>2 INFORMATION: PGM IEFBR14 FOUND IN LINKLIST LIBRARY SYS1.LINKLIB</td>
</tr>
<tr>
<td></td>
<td>3 //NEWTEMP DD SPACE=(TRK,1),DSN=NEW.FILE.NO.DISP</td>
</tr>
<tr>
<td></td>
<td>4 //FILEOK DD DISP=SHR,DSN=CTJP.V800.JCL.SAMPLES</td>
</tr>
<tr>
<td></td>
<td>5 //</td>
</tr>
<tr>
<td>CTJ009I</td>
<td>Processing ended RC=0000 for job IEFBR14 member IEFBR14</td>
</tr>
<tr>
<td>CTJ003I</td>
<td>Processing ended RC=0000 REASON 00000000 member IEFBR14 DSNAME CTJP.V800.JCL.SAMPLES</td>
</tr>
<tr>
<td>CTJU02I</td>
<td>Control-M/JCL Verify Utility ended. RC=0000</td>
</tr>
</tbody>
</table>

Example 5

The following example shows a CTJVER output report for a job, which if submitted, would be rejected by the system because of a JCL error.

When a JCL error is found, Control-M/JCL Verify does not perform the second verification phase because the input might be invalid. Figure 9 shows the job that is to be verified.
Figure 9  Verified job - Sample01 – Example 5

```
//JCLERROR JOB ,BR14,MSGCLASS=X,NOTIFY=N18A,MSGLEVEL=(1,1)  
//S1   EXEC PGM=IEFBR14,COND=NEVER
//DSNM  DD DISP=SHR,DSM=CTJP.V800.JCL.SAMPLES(JCLERR)
//DISP=  DD SPACE=(TRK,1),DSN=N18.SYSTEM.NO.0S35
```

Figure 10 shows the job that performs the verification.

Figure 10  CTJVER utility – Example 5

```
//SAMPLE01 JOB 0,YY,CLASS=A,MSGCLASS=X.  
//     NOTIFY=&SYSUID
//   JCLLIB ORDER=IOAP.V800.PROCLIB
//  INCLUDE MEMBER=IOASET
//SAMPLE01 EXEC CTJVER
//SYSIN DD *
LIB=CTJP.V800.JCL.SAMPLES MEM=JCLERR1
/*
```

As shown in Figure 11, IBM informational messages, usually starting with an IEFC prefix, are displayed similar to how they appear in the JOB JESJCL output. Control-M/JCL Verify marks the IBM messages with $JCLERR>$ at the beginning of the line.

If the job has a JCL error, or contains any error that requires changing the JCL statements, the validation ends with return code (RC) 8.

Figure 11  CTJVER utility - SYSPRINT – Example 5

```
CTJU01I  FA=D FE=D JES=D ME=D ML=D
CTJU01I  LIB=CTJP.V800.JCL.SAMPLES MEM=JCLERR
CTJ002I Start verifying member JCLERR DSNAME CTJP.V800.JCL.SAMPLES
CTJ008I Start verifying JOB JCLERROR MEMBER JCLERR
    1 //JCLERROR JOB ,BR14,MSGCLASS=X,NOTIFY=N18A,MSGLEVEL=(1,1)
    2 //S1   EXEC PGM=IEFBR14,COND=NEVER
    3 //DSNM  DD DISP=SHR,DSM=CTJP.V800.JCL.SAMPLES(JCLERR)
JCLERR>  3 IEFC630I UNIDENTIFIED KEYWORD DSM
    4 //DISP=  DD SPACE=(TRK,1),DSN=N18.SYSTEM.NO.0S35
JCLERR>  4 IEFC662I INVALID LABEL
    5 //
CTJ009I Processing ended RC=0008 for job JCLERROR member JCLERR
CTJ003I Processing ended RC=0008 REASON 00000000 member JCLERR DSNAME CTJP.V800.JCL.SAMPLES
CTJU02I Control-M/JCL Verify Utility ended. RC=0008
```

Example 6

The following example shows a CTJVER output report for a job that is to be submitted by Control-M monitor.

The job is part of SMART table SAMPLE08.
Figure 12 shows the relevant parts of the SMART Table Entity Definition in Screen 2. The SMART Table contains variable that %%EX_SYSTEM will be set in the JCL.

Figure 12  SMART Table with variable %%EX_SYSTEM – Example 6

---

Figure 13 shows the relevant parts of the JOB Definition SAMPLE08 in Screen 2. The JOB definition table contains variable that %%EXDATE will be set in the JCL.

Figure 13  Job definition with variable %%EXDATE – Example 6
Figure 14 shows JCL of the job specified in the SAMPLE08 definition above. The job contains three statements that are resolved when Control-M monitor submits the job:

- /*ROUTE PRINT %%EX_SYSTEM - A JES2 statement that sends the printout to a specific z/OS system.
- /* SET %%SAMPLE08_OLD=%%$CALCDTE %%$ODATE -1 - A Control-M statement to resolve the day of the day before the run
- //NEWFILE DD DISP=(DELETE),DSN=CTJP.SAMPLE08.D%%EXDATE, - A JCL statement that its DSN last level contains the current date.

Figure 15 shows the job that performs the verification.
The CTJVER utility output report, shown in Figure 16, includes various messages. To simplify the reading and interpretation of the messages, Control-M/JCL Verify combines the messages that are issued by the system in the JESYSMSG member with the JCL statements.

Figure 16  CTJVER utility - SYSPRINT – Example 6

In the above sample:

- CTJ002I - shows from which library the JCL member was read

Note: if the MEMLIB is a DD name like GENERAL, the library name is the first library in the DD statement concatenation.
CTJR05I - echoes the original record in the JCL member that contains Control-M AutoEdit Variables or IOA Global AutoEdit Variables. This record is replaced by the JCL statement appearing in the next record, which follows it in the SYSPRINT output. This message is also displayed if Control/M/JCL verify changes a JCL statement.

Example 7

The following example shows a CTJVER output report for the job in SAMPLE08 (see “Example 6” on page 39), which was ordered and is now in the Control-M AJF. The orderid of the job is 020QM.

Figure 17 shows the job that performs the verification.

Figure 17  Verified job - Sample01 – Example 7

The CTJVER utility output report, shown in Figure 18, includes various messages. To simplify the reading and interpretation of the messages, Control-M/JCL Verify combines the messages that are issued by the system in the JESYSMSG member with the JCL statements.

Figure 18  CTJVER utility - SYSPRINT – Example 7 (part 1 of 2)
In the above sample, the values of the resolved variables are taken from the Control-M AJF records and they are the values that Control-M will use when the job is submitted.

Example 8

In the following example, CTJVER verifies the control statements in IEBGENER.

The required input and output DD files are:

- **SYSPRINT** - The BLKSIZE value can be any value up to 32670 that is a multiple of 121 and a RECFM of F or FB.
Example 9

In the following example, CTJVER verifies the control statements in SORT.

Sort has several aliases that can be used in JCL.

Figure 20  SORT – Example 9

```
//K68TALL  JOB ,K68,MSGCLASS=X,CLASS=A,
//         NOTIFY=N18
//@
//@ SORT      EXEC  PGM=SORT PARM='MSGDDN=MSGDD'
//@
//@SORTIN    DD DISP=SHR,DSN=CTJP.SORT.CTMDAILY.SYSOUT
//@SORTOUT   DD SYSOUT***
//@SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//@SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//@SYSOUT    DD SYSOUT***
//@MSGDD    DD SYSOUT***
//@DFSPARM  DD *
//@OPTION MSGDDN=MSGDD
//@SYSIN    DD *
//@       FIELDS=COPY
//@       INCLUDE COND=(11,7,CH,EQ,C"JOBS501I",AND,19,1,CH,NE,C"**")
//@       RECORD TYPE=F
//@       OPTION DYNALLOC=SYSDA
//@       OUTFIL OUTREC=(18,97)
//@       OPTION NOOPTIONS
```
Example 10

In the following example, CTJVER verifies the control statements in IEBCOPY.

The required input and output DD files are:

- **SYSPRINT** - The **BLKSIZE** value can be any value up to 32670 that is a multiple of 121 and a **RECFM** of F or FB.

- **SYSIN** - The **BLKSIZE** must be a multiple of 80, with a maximum allowed value of 32,720, and a **RECFM** of F or FB.

If **SYSIN** is a dummy file, **SYSUT1** or **SYSUT2**, or both, might be required, depending on the parameters in the **EXEC** statement.

**Figure 21  IEBCOPY – Example 10**

```
//K6BTALL   JOB ,K6B,MSGCLASS=X,CLASS=A,
//         NOTIFY=N18
//*
//IEBCOPY  JOB (ACCOUNT),'IEBCOPY'
//*
/COPYJOBS EXEC PGM=IEBCOPY,PARM='LIST=NO'
//SYSPRINT DD SYSOUT=*  
//RESOURCE DD DISP=SHR,DSN=QUEST.JVER.INPUTA
//BACKUP   DD DISP=(,CATLG),DSN=QUEST.JVER.INPUTA,
//         LRECL=80,BLKSIZE=32000,RECFM=FB,SPACE=(CYL,(1,1,10))
//SYSIN    DD *
COPY001 COPY O=OUTX,I=INA
```
The site standards feature ensures that JCL statements comply with required enterprise standards.

This chapter includes the following topics:

- Introduction ................................................................. 49
- Overview ................................................................. 49
- Rule selection and searching ........................................ 49
- DO block ................................................................. 50
- Variables ................................................................. 51
- Defining site standard rules online .................................. 53
- Loading site standard rules ........................................... 60
- Verifying site standards using REXX EXECs ....................... 61

Rule Definition Facility .................................................. 63
- Entry Panel ............................................................. 66
- Table List Screen ......................................................... 69
- Rule List Screen .......................................................... 71
- Copying Rules to Another Table .................................... 74
- Rule Definition Screen .................................................. 76
- Commands of the Rule Definition Screen ......................... 82
- Editing Rule Definitions in the Edit Environment ............... 82
- Exiting the Rule List Screen .......................................... 84
- Copying Rules to Another Table .................................... 85

Parameter Descriptions .................................................. 87
- CONTINUE SEARCH: General Parameter ......................... 88
- DESC: General Parameter ............................................... 89
- DO statement: Action Parameter ..................................... 90
- DO IF / DO ELSE / DO ENDIF: Action Parameter ............... 92
- DO MSG: Action Parameter ............................................. 95
- DO RETURN: Action Parameter ....................................... 96
- DO REXX: Action Parameter ........................................... 97
- DO SET: Action Parameter .............................................. 98
- ENVIRONMENT: General Parameter ................................. 107
- JOBNAME: Common Selection Parameter ......................... 109
- LIBRARY: Common Selection Parameter ............................ 110
MEMBER: Common Selection Parameter ........................................... 111
ON DD: Event Selection Parameter .................................................. 112
ON DDSYSIN: Event Selection Parameter ......................................... 115
ON DDSYSOUT: Event Selection Parameter ...................................... 116
ON EXEC: Event Selection Parameter .............................................. 117
ON INCLUDE: Event Selection Parameter ......................................... 119
ON JCLLIB: Event Selection Parameter ........................................... 120
ON JOB: Event Selection Parameter .............................................. 121
ON JOBEND: Event Selection Parameter .......................................... 123
ON JOBINIT: Event Selection Parameter ......................................... 124
ON PROC: Event Selection Parameter ............................................ 125
RULE: General Parameter ............................................................... 126
SCHDLIB: Common Selection Parameter ......................................... 127
SCHDTAB: Common Selection Parameter ......................................... 128
Introduction

The site standards feature ensures that JCL statements comply with required enterprise standards.

Overview

To test for compliance to enterprise standards, the user defines rules that incorporates the site standards and runs Control-M/JCL Verify to test whether the JCL statements actual comply with the standards.

The site standard rules are based on an event-action triggering mechanism using ON (for detecting events) and DO (for performing actions) statements. This mechanism is similar to the one which is already implemented in other IOA products such as Control-M (CME), Control-M/Tape, and Control-O.

For each site standard rule the user defines one or more

- ON statements - to specify what will trigger the test
- DO statements - to specify what is tested and what actions are taken based on the results of the test

There are several available types of ON statements, described later in the chapter, each of which handles a specific JCL statement type (for example, JOB, EXEC, and DD) or other JCL parameter.

ON statements have a set of one or more selection fields for matching the appropriate rule to the JCL statements requiring testing.

Rule selection and searching

Rule selection fields

The rules which are used in the job verification are selected based on the following selection fields:

- environment - can be optionally assigned to rules, limiting the site standard verification to only those rules in which the particular environment name is specified.
- common selection fields - common to all the various ON statements, such as JOBNAME, LIBRARY, and MEMBER.

- event selection fields - specific to particular ON statements, such as CLASS, USER, and ACCOUNT.

The wild card characters, such as the masks using asterisk (*) and the question mark (?), which are commonly used in IOA, can be used in the selection fields.

If a selection field is left blank it is equivalent to specifying an asterisk, and that field will be eliminated from the selection process.

**Rule search criteria**

When a JCL statement requires testing, a search is performed to locate the appropriate site standard rule. Since different rules can be defined with similar selection fields, it is possible that more than one rule will match a JCL statement. Therefore, rules are searched and used according to the criteria in the following order:

- priority - Each rule is defined with a priority from 1 to 99. The rules with the higher priority (larger number) are searched first. The default priority is 50.

- level of detail - Rules in which the selection fields are defined with more specific details are searched first. For example, DSN=A.B.* is more specific than DSN=A.* when a dataset with the name of A.B.C is processed.

- continuity - Each rule is defined with a setting that determines whether to continue searching for additional rules that meet the selection criteria. A setting of N (No) stops the search for additional rules.

**DO block**

Each rule contains a DO block to perform the site standards validation.

This block supports a set of commands and built-in functions to perform the validation. The data, from JCL statements or external resources (such as member name, library name, and environment) is stored in variables by the system. The user can also define temporary variables and store data in them.
Variables

Variables are data objects which are available to the user in the rules. The user can keep data in them in order to pass information from one rule to another.

The following types of variables are available:

■ general system variables
■ card data variables
■ user-defined variables

The following types of variables are available:

General system variables

General system variables are variables that get their value by JVER when the site standards verification begins. The user cannot update them using the SET command. All the system variables begin with %%# to differentiate them from the other variable types.

The system variables are available to all the rules. The system variables are described in Table 4. The date system variables are described in Table 5.

Table 4 System variables (part 1 of 2)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%%#JCLTYPE</td>
<td>The type of JCL statement (for example, DD, EXEC, and JOB).</td>
</tr>
<tr>
<td>%%#MEMBER</td>
<td>The JCL statements member name.</td>
</tr>
<tr>
<td>%%#LIBRARY</td>
<td>The library name containing the JCL member.</td>
</tr>
<tr>
<td>%%#ENVRN</td>
<td>The environment specified in the invocation.</td>
</tr>
<tr>
<td>%%#SCHDLIB</td>
<td>The scheduling library with the job definition (if verifying Control-M job definition).</td>
</tr>
<tr>
<td>%%#SCHDTAB</td>
<td>The scheduling table name (if verifying Control-M job definition)</td>
</tr>
<tr>
<td>%%#SCHDJOB</td>
<td>The scheduling job name (if verifying Control-M job definition) it will be equal to %%#MEMBER.</td>
</tr>
<tr>
<td>%%#STATNUM</td>
<td>The statement number (after the procedures and Control-M cards are added). During JOBINIT the value of %%#STATNUM is set to zero.</td>
</tr>
<tr>
<td>%%#DDJOB</td>
<td>The number of DD in job.</td>
</tr>
<tr>
<td>%%#STPJOB</td>
<td>The number of steps in job.</td>
</tr>
<tr>
<td>%%#LINJOB</td>
<td>The number of lines/records in job.</td>
</tr>
</tbody>
</table>
### Table 4  System variables (part 2 of 2)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%%%#DDSTP</td>
<td>The number of DDs in step.</td>
</tr>
<tr>
<td>%%%$SRC</td>
<td>The highest return code attained up until the current point in the procedure. The format is: \text{RETURN RC 0000 RS X'0000}</td>
</tr>
<tr>
<td>%%%$RS</td>
<td>The highest reason code attained up until the current point in the procedure. Expressed in HEX code. The format is: \text{RETURN RC 0000 RS X'0000}</td>
</tr>
</tbody>
</table>

### Table 5  Date system variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%%%$DATE</td>
<td>The current system date, in the format yymmdd.</td>
</tr>
<tr>
<td>%%%$ODATE</td>
<td>The current ODATE (when invoked from a Control-M interface). The current ODATE (when invoked from a Control-M interface).</td>
</tr>
<tr>
<td>%%%$CENT</td>
<td>The first two digits in the current year</td>
</tr>
<tr>
<td>%%%$DAY</td>
<td>The current system day, in \text{dd} format.</td>
</tr>
<tr>
<td>%%%$JULDAY</td>
<td>The current system day, in \text{jjj} format.</td>
</tr>
<tr>
<td>%%%$MONTH</td>
<td>The current system month, in \text{mm} format.</td>
</tr>
<tr>
<td>%%%$RDATE</td>
<td>The installation current working date, in \text{yymmdd} format.</td>
</tr>
<tr>
<td>%%%$RDAY</td>
<td>The installation current working day, in \text{dd} format.</td>
</tr>
<tr>
<td>%%%$RJULDAY</td>
<td>The installation current working day of the year, in \text{jjj} Julian format.</td>
</tr>
<tr>
<td>%%%$RMONTH</td>
<td>The installation current working month, in \text{mm} format.</td>
</tr>
<tr>
<td>%%%$RWDAY</td>
<td>The current working day of the week, in \text{d} format, where \text{d} is 1 through 6 or 0. For example, 1=Sunday, 2=Monday, ...6=Friday, 0=Saturday.</td>
</tr>
<tr>
<td>%%%$RYEAR</td>
<td>The installation current working year, in \text{yy} format.</td>
</tr>
<tr>
<td>%%%$TIME</td>
<td>The time of day, in \text{hhmmss} format.</td>
</tr>
<tr>
<td>%%%$WDAY</td>
<td>The current Gregorian day of the week in \text{d} format, where \text{d} is from 1 through 6 or 0. For example, 1=Sunday, 2=Monday, ...6=Friday, 0=Saturday.</td>
</tr>
<tr>
<td>%%%$YEAR</td>
<td>The current system year, in \text{yy} format.</td>
</tr>
<tr>
<td>%%%$JCL#</td>
<td>Total number of JCL statements of the final JCL. 0 is not valid</td>
</tr>
<tr>
<td>%%%$JCLn</td>
<td>The nth JCL statement, up to 72 characters.</td>
</tr>
</tbody>
</table>
Card data variables

Data variables are variables that contain data from the JCL card. For each JCL statement, there is a set of predefined variables representing the parameters of the specific JCL statement type. All the card data variables begin with %%$ to differentiate them from the user variables. The variables that are available with each JCL card are discussed in the relevant ON statement for each JCL card type later in the chapter.

Card data variables are available in the rule that processed them and in all the following rules until the same card is processed (and then the variables are set with the new card data).

If the data variable is not referenced in the card processed, the variable value would be set to null.

For example, the data from the JOB card is available to all the following cards in the job. The data from the EXEC card is available to all the DD cards that follow it until it is replaced with the next EXEC card data (which in turn would be available for all the DD cards that follow it).

User-defined variables

These variables are defined by the SET commands that are included in the DO section of the rules. The variable names begin with %%. The variables are available to all the rules used during the JCL verification and are deleted when the job verification ends.

User defined variables are available to all the rules that follow the rule that define them (in the first rule that has a SET command for the variable). For example, a variable defined by a SET command that processed the EXEC cards is available to all the DD cards that follow it, to all the EXEC cards that follow it, but not to the JOB cards that precede it.

The scope of user defined variables are on a “job - level.” The variables are deleted after the job verifications are completed.

The length of user local variables for the site standard rules is limited to 12 characters, including the %% characters at the beginning.

Defining site standard rules online

The selection criteria and instructions defined for each site standard rule are called rule parameters. This section briefly describes the rule parameters and their specific uses. For more information about the specific parameters, see “Parameter Descriptions” on page 87.
The site standard rule parameters are defined in the Rule Definition screen, shown in Figure 22. This is the main screen of the Rule Definition facility. This section briefly describes the Rule Definition screen. For more information about the Rule Definition facility, see “Rule Definition Facility” on page 63.

**Figure 22  Rule Definition Screen**

<table>
<thead>
<tr>
<th>RULE</th>
<th>ENVIRONMENT</th>
<th>PRIORITY</th>
<th>CONTINUE SEARCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESC</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**ON Types:**

- JOBINIT
- JOBEND
- JOB
- INCLUDE
- PROC
- DD
- DDSYSIN
- DDSYSOUT

---

**DO Types:**

- IF
- ELSE
- RETURN
- ENDF
- SET
- MSG
- REXX

---

FILL IN RULE DEFINITION. CMDS: EDIT DOOPT ONOPT CANCEL 16.38.45

The rule panel is divided into the following sections:

- **identification fields**
  
  This section includes the rule name and its description. These fields are used for display purposes only, but have no effect on the rule logic. This section also includes the environment field, which is used in the rule selection criteria, the rule priority, and the continue search indicator.

- **common selection fields**
  
  This section includes the selection fields that are common to all the various ON types. Several selection blocks can be specified with Or/Not relationships.

- **ON definition**
This section includes the ON statement. The ON defines the reason for the rule invocation. In most of the cases the reason is the processing of a JCL statement. The section can hold more than one ON statement (an empty row is opened once the preceding one is used – similar to other IOA panels). Once the ON statement is specified, the specific ON selection fields (if any) is added to the panel. Refer to the rules selection fields section above. The specific selection fields for each ON statement are discussed in the relevant ON section later in this document. Several selection blocks for the ON specific fields can be specified with Or/Not relationships.

■ available ON statements

This section simply lists the available ON statements from which the user can choose. This section can be hidden using the ONOPT command (or the short ON command), which allows the user to toggle between the hidden and displayed states of the section.

■ DO definition

This section includes the DO statement. The DO defines what action will be performed if the rule is triggered. The rule is triggered when the ON “event” has occurred and all the rule’s selection fields match. The different DO statements are described in detail later in this document. The section can hold more than one DO statement (an empty row is opened once the preceding one is used).

■ available DO statements

This section simply lists the available DO statements from which the user can choose. This section can be hidden using the DOOPT command, which allows the user to toggle between the hidden and displayed states of the section.

All parameter definitions are stored as members in $JVER operation libraries prefix.RULES library. The Control-M JCL Verify term for a member of this type is “table.” Each table is composed of parameters for a number of different rules, all of which usually relate to the same subject. Maintenance of tables in the library and management of rules in each table is performed using the Online Rule Definition facility.

Rule definition parameters fall into the following basic categories:

■ General Parameters – general information.

■ Common Selection Parameters – common selection conditions.

■ Event Selection Parameters – event selection conditions.

■ Action Parameters – actions to be performed.
The following pages contain a quick summary of the parameters in each of these categories. Following this summary is a series of sample rules that demonstrate how basic tasks are accomplished.

The last (and largest) portion of this chapter contains detailed descriptions of all rule definition parameters (in alphabetical order).

**General Parameters – Summary**

General parameters provide general information about the rule and to provide certain criteria for rule selection. The General parameters are shown in Table 6.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RULE</td>
<td>Name of a rule as specified in the entry panel.</td>
</tr>
<tr>
<td>ENVIRONMENT</td>
<td>Name of the environment where the rule applies.</td>
</tr>
<tr>
<td>PRIORITY</td>
<td>Specifies the priority of the rule.</td>
</tr>
<tr>
<td>CONTINUE SEARCH</td>
<td>Specifies whether additional rules will be searched and used.</td>
</tr>
<tr>
<td>DESC</td>
<td>Free-text description of the rule definition.</td>
</tr>
</tbody>
</table>

**Common Selection Parameters – Summary**

Common Selection parameters, which are common to all the various ON parameters types (see “Event Selection Parameters – Summary” on page 57), specify which JCL statements will trigger the validation processes to be performed by Control-M JCL Verify. Several selection blocks can be specified with Or/Not relationships. The Common Selection parameters are shown in Table 7.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBNAME</td>
<td>Name of the job (as appears in the JOB statement).</td>
</tr>
<tr>
<td>LIBRARY</td>
<td>Name of the library where the JCLs reside. (Note that this is also the MEMLIB, if invoked from a Control-M/EM interface.)</td>
</tr>
<tr>
<td>MEMBER</td>
<td>Name of the member name in the library. (Note that this is also the MEMNAME, if invoked from a Control-M/EM interface.)</td>
</tr>
<tr>
<td>SCHDLIB</td>
<td>Name of the scheduling library when invoked from a Control-M/EM interface.</td>
</tr>
</tbody>
</table>
Event Selection Parameters – Summary

Event Selection parameters (ON parameters) specify which job events or JCL statements will trigger the validation processes to be performed by Control-M JCL Verify. Valid ON parameters are shown in Table 8.

Table 7 Common Selection Parameters (part 2 of 2)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCHDTAB</td>
<td>Name of the scheduling table when invoked from a Control-M/EM interface. (Note that the name does not refer to “folder” since “table” is still the term used with Control-M for z/OS.)</td>
</tr>
<tr>
<td>O/N</td>
<td>Conjunctional subparameter that permits linking of common selection statements. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>O (Or) – Only one statement need be satisfied.</td>
</tr>
<tr>
<td></td>
<td>N (Not) – The condition in the following statement need not be satisfied.</td>
</tr>
</tbody>
</table>

Table 8 Event Selection Parameters (ON Parameters) (part 1 of 2)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON JOBINIT</td>
<td>Specifies that the rule is triggered when a JCL job is about to be processed. Used to initialize user variables (for example, counters).</td>
</tr>
<tr>
<td>ON JOB</td>
<td>Specifies that the rule is triggered by a JOB statement. Used to validate the JOB statement.</td>
</tr>
<tr>
<td>ON EXEC</td>
<td>Specifies that the rule is triggered by an EXEC statement. Used to validate the EXEC statement.</td>
</tr>
<tr>
<td>ON DD</td>
<td>Specifies that the rule is triggered by a DD statement. Used to validate the DD statement.</td>
</tr>
<tr>
<td>ON JOBEND</td>
<td>Specifies that the rule is triggered when a JCL job is ending. Used, for example, for cross JCL validations and counter checking.</td>
</tr>
<tr>
<td>ON INCLUDE</td>
<td>Specifies that the rule is triggered by an INCLUDE statement. Used to validate the INCLUDE statements.</td>
</tr>
<tr>
<td>ON PROC</td>
<td>Specifies that the rule is triggered by a PROC statement. Used to validate the PROC statements.</td>
</tr>
<tr>
<td>ON DDSYSIN</td>
<td>Specifies that the rule is triggered by a SYSIN DD statement (either DD * or DD DATA). Used to validate the SYSIN DD statements.</td>
</tr>
</tbody>
</table>
When you specify an ON parameter and press Enter, the subparameters for this ON parameter are displayed. The combination of an ON parameter and its subparameters is called an ON statement.

Multiple ON statements, of the same type, can be specified in a rule. The Or/Not conjunctional parameter is used to link two or more ON statements.

For a general explanation of event selection parameters, see “ON DD: Event Selection Parameter” on page 112.

### Action Parameters – Summary

Action parameters (DO parameters) specify actions to be performed by Control-M JCL Verify. These actions are performed only after conditions specified in the Event Selection parameters have been fulfilled. Note that the Boolean “IF” logic capabilities are provided using the DO IF, DO ELSE, or DO ENDIF statements.

The Action parameters are shown in Table 9.

### Table 8 Event Selection Parameters (ON Parameters) (part 2 of 2)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON JCLLIB</td>
<td>Specifies that the rule is triggered by a JCLLIB statement. Used to validate the JCLLIB statements.</td>
</tr>
<tr>
<td>ON DDSYSOUT</td>
<td>Specifies that the rule is triggered by a SYSOUT DD statement set to a specific value. Used to validate the DD statement with SYSOUT equal to the specified value.</td>
</tr>
</tbody>
</table>

### Table 9 Action Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DO IF</td>
<td>Provide Boolean “IF” logic capability allowing for alternative actions to be performed.</td>
</tr>
<tr>
<td>DO ELSE</td>
<td>Exit from the rule and returns to Control-M JCL Verify.</td>
</tr>
<tr>
<td>DO ENDIF</td>
<td>Call user exit written in REXX.</td>
</tr>
<tr>
<td>DO SET</td>
<td>Assign a value to an AutoEdit variable.</td>
</tr>
<tr>
<td>DO MSG</td>
<td>Define a message to be displayed in the validation output.</td>
</tr>
</tbody>
</table>

For a general explanation of automated console action parameters, see “DO statement: Action Parameter” on page 90.
The following pages contain detailed descriptions of all parameters available in the Rule Definition screen. Parameters are arranged in alphabetical order. Within each parameter, subparameters are arranged according to the order of the fields on the screen.

Each parameter begins on a new page, including:

- a brief explanation of the purpose of the parameter
- the format required for defining the parameter within an extract of the Control-O screen
- general information explaining the parameter and its usage
- where applicable, some practical examples illustrating implementation of the parameter

For more information on the Rule Definition facility, see “Rule Definition Facility” on page 63.
The CTJINIT utility is used to load or remove the site standard rules listed in the CTJRULE member, located in the $olprefj.PARM library. The CTJINIT utility can be run as a started task.

The operator can perform the various CTJINIT functions by using the following commands:

**To load the site standard rules**

- S ctjJREF,SCOPE=RULES
- S ctjJREF,SCOPE=ALL

**To remove the site standard rules**

- S ctjJREM,SCOPE=RULES
- S ctjJREM,SCOPE=ALL

**To forcefully remove the site standard rules**

- S ctjJREMF,SCOPE=RULES
- S ctjJREMF,SCOPE=ALL

where ctj are the first three characters of the Control-M/JCL Verify JCL procedures defined in PROCPRFJ

The SCOPE parameter determines which data is loaded or removed. When SCOPE is set to ALL, the loading or removing function is applied to all Control-M/JCL Verify tables and the rules listed in the CTJRULE member. When SCOPE is set to RULES, the loading or removing function is applied to only to the rules listed in the CTJRULE member.

For more information, see Chapter 9, “CTJINIT utility.”
Verifying site standards using REXX EXECs

The user can call REXX EXECs, and after the site standards is activated, Control-M/JCL Verify functions and variables can be used in the REXX EXEC for verification of site standards.

Call format

The following formats can be used to call the REXX EXEC for site standard verification:

- Function call

  \[ \text{var} = \text{CTJRXX}("function","%%variable = value") \]

- ADDRESS

  \[ \text{ADDRESS JVER("function","%%variable = value")} \]

**NOTE**

The function call format is mandatory for the initial call CTJRXX("INIT"), since it sets up the support for ADDRESS JVER format. After the initialization, either format can be used.

<table>
<thead>
<tr>
<th>Table 10</th>
<th>REXX EXEC call functions (part 1 of 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Function</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>INIT</td>
<td>Initiates the JVER environment under the REXX environment.</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
</tr>
<tr>
<td></td>
<td>( I = \text{CTJRXX}(&quot;INIT&quot;) )</td>
</tr>
<tr>
<td>SETVAR</td>
<td>Sets local variables, which can be used later in other DO REXX actions or even in any other DO action.</td>
</tr>
<tr>
<td></td>
<td>Formats:</td>
</tr>
<tr>
<td></td>
<td>( S = \text{CTJRXX}(&quot;SETVAR&quot;,&quot;%%A = B&quot;) )</td>
</tr>
<tr>
<td></td>
<td>ADDRESS JVER &quot;SETVAR %%A = B&quot;</td>
</tr>
</tbody>
</table>
Table 10  REXX EXEC call functions (part 2 of 2)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOMSG</td>
<td>Issues a JVER message in the JVER report, similar to DO MSG in the site standards.</td>
</tr>
<tr>
<td></td>
<td>Formats:</td>
</tr>
<tr>
<td></td>
<td>M = CTJRXX(&quot;DOMSG&quot;,&quot;I&quot;,&quot;&lt;text&gt;&quot;)</td>
</tr>
<tr>
<td></td>
<td>ADDRESS JVER &quot;DOMSG I\WE &lt;text&gt;&quot;</td>
</tr>
<tr>
<td>DORC</td>
<td>Sets the JVER return code (RC) and reason code (RS) of the JVER report, similar to DO RETURN in the site standards.</td>
</tr>
<tr>
<td></td>
<td>Formats:</td>
</tr>
<tr>
<td></td>
<td>R = CTJRXX(&quot;DORC&quot;,&quot;8&quot;,&quot;0&quot;)</td>
</tr>
<tr>
<td></td>
<td>ADDRESS JVER &quot;DORC 8 A&quot;</td>
</tr>
<tr>
<td>RESOLVE</td>
<td>Resolves a local, system, or data variable, and gets its value.</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
</tr>
<tr>
<td></td>
<td>A = CTJRXX(&quot;RESOLVE&quot;,&quot;%%A&quot;)</td>
</tr>
</tbody>
</table>

Variables

Besides local, system, and data variables, the following system variables for site standards are available:

- %%%$JCLn – The nth JCL statement, up to 72 characters.
- %%%$JCL# – Number of JCL statements of the final JCL.

Example

In the following example, REXX user exit is initiated and then the jobname is retrieved with the RESOLVE function. When the jobname is K68 the following message is printed out: INFORMATION: THE JOBNAME IS K68.

Figure 23  Example of REXX call

```/* REXX */
I = CTJRXX("INIT")
A = CTJRXX("RESOLVE","%%$JOBNAME")
IF A = "K68" THEN DO
  M = CTJRXX("DOMSG","I","THE JOBNAME IS K68")
```
Rule Definition Facility

The Control-M/JCL Verify Rule Definition facility is composed of a series of screens that enable you to view, create, modify, and delete rule definitions and parameters.

Rule definitions under Control-M/JCL Verify are stored in the JVER operation libraries prefix.RULES. library. The library contains rule tables (members), and each table contains rules (criteria) for one or more console messages, commands, strings, or events. A table generally contains rules related to a specific topic, such as IPL.

--- NOTE ---

The Control-M/JCL Verify Rule Definition facility does not support members which have been compressed using the ISPF PACK option.

The number of rule tables in the library, the number of rules in a table, and the size of each rule definition, are all calculated dynamically and are not dependent on parameter specification.

Accessing the Control-M/JCL Verify Rule Definition Facility

The Control-M/JCL Verify Rule Definition facility contains the screens shown in Table 11.

<table>
<thead>
<tr>
<th>Screen</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control-M/JCL Verify Rule Definition Entry Panel</td>
<td>Allows specification of parameters which determine which screen is displayed.</td>
</tr>
<tr>
<td>Control-M JCL Verify Rules - Table List screen</td>
<td>Displays the list of tables (members) in the Control-M/JCL Verify rule library.</td>
</tr>
<tr>
<td>Control-M JCL Verify Rules - Rules of Table List screen</td>
<td>Displays the list of rules in a specific rule table.</td>
</tr>
<tr>
<td>Control-M/JCL Verify Rule (definition) screen</td>
<td>Displays the parameters of the selected rule. This screen is used to define, display and modify rule parameters.</td>
</tr>
</tbody>
</table>
To enter the Rule Definition facility, select Option JR on the IOA Primary Option menu and press Enter. The entry panel is displayed.

Creating Tables

Tables are created by specifying the new table name in the entry panel and pressing Enter. As a result, a skeletal rule definition (that is, one with most fields left blank) is displayed in the rule definition screen.

Fill in the blanks and save the screen entries. The table is created and the rule definition is the first and only definition in the Rule list of the table. As additional rule definitions are created in the table (described below), they are added to the list.

--- NOTE ---
Upon exiting the Rule List screen, if changes were made in at least one rule definition, an Exit Option window is displayed. One field of the window displays the table name. This value can be changed to a new table name that creates a new table in which the rule definitions are saved.

Creating Rule Definitions

Rule definitions can be created using two basic methods

- A skeletal rule definition can be created by specifying the name of a new rule definition in the entry panel. The table specified in the entry panel can be either a new or an existing table. In this case, virtually all fields of the rule definition are empty.

- A copy of an existing rule definition can be created using the INSERT option (described in “Options of the Rule List Screen” on page 73) in the Rule List (Rules of Table: ABC) screen. In this case, most fields of the new rule definition have the same values as the fields in the copied rule definition.

Performing Operations on Tables and Rules

Many operations can be performed on tables and on the rule definitions in them. These operations are performed using commands and options in the various screens of the Rule Definition facility.

Following is a brief summary of some of the major operations possible within the facility. Additional options and commands are explained following the summary.
Accessing (Editing or Browsing) a Table and its Rules

A table (that is, the rule definitions in the table) can be browsed or edited.

When browsed, the table cannot be modified or updated. When the table is edited, new rule definitions can be added and existing rule definitions can be modified or deleted.

Browsing, however, has the following advantages:

- Access and exit are quicker than in editing.
- Rule lists and definitions that are in use by another user can be viewed.
- Access for browsing might be granted, while though access for editing might be denied due to site security requirements.

To browse a table (and its rule list and rule definitions), use the BROWSE option in the Table List screen.

Specifying the table name in the entry panel or using the SELECT option in the Table List screen provides edit access.

Depending on user profile definitions, if the table requested for editing is in use, access is either granted in Browse mode, or access will not be granted.

Copying a Rule to Another Table

Rules can be copied from one table to another using the COPY option in the Rule List screen. For more information, see “Copying Rules to Another Table” on page 85.

Deleting a Table or a Rule

Unneeded rules can be deleted using the DELETE option in the Rule List screen. For more information, see “Options of the Rule List Screen” on page 73. Unneeded tables can be deleted using the DELETE option in the Table List screen. For more information, see “Deleting Tables” on page 70.

Saving Modifications

All changes made to a table and its rule scheduling definitions are kept in memory until the table is exited. Upon exiting the table, the user can choose to save or cancel the changes. For more information, see “Exiting the Rule Definition Screen” on page 83.
Entry Panel

Overview

The entry panel is displayed upon entering the Rule Definition facility (Option JR in the IOA Primary Option menu).

Figure 24  Rule Definition Facility Entry Panel

<table>
<thead>
<tr>
<th>Command</th>
<th>Control-M/JCL Verify Rule Definition Entry Panel (JR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>(Blank for table selection list)</td>
</tr>
<tr>
<td>Rule</td>
<td>(Blank for rule selection list)</td>
</tr>
</tbody>
</table>

Fill in entry panel fields TABLE and RULE, as described below, to open the desired display.

**NOTE**

If you use the selection list fields, their values are not erased until you exit the entry panel by pressing END (PF03/PF15).

- To display the list of rules for a specific table, do the following:
  1. Type the table name.
  2. Press Enter.

  If the table does not exist, the screen for defining a new rule is displayed.

- To display the details of a specific rule (Rule Definition screen), do the following:
1. Type the table name.

2. Type the rule name.

3. Press Enter.

If the table does not exist, or the rule in the specified table does not exist, the screen for defining a new rule in the table is displayed.

---

**NOTE**

If you enter the screen for defining a new rule and want to leave the screen without defining a rule, use the CANCEL command.

---

- To display the Search Window (described below), do the following:

  1. Type the rule name.

  2. Leave the table name blank or type the table name using mask characters (* or ?).

  3. Press Enter.

- To create a new rule table, do the following:

  1. Specify a new table name.

  2. Press Enter.

  The Rule Definition screen, for defining the first rule in the new table, is displayed.

**Search Window**

The Search window, illustrated in Figure 25, allows the user to search for the specified rule in tables of the specified library. Tables in which the rule has been found are then displayed in the Table List screen.
To close the Search Window without performing any action, press PF03/PF15 (END).

To perform a search, select one of the following choices and press Enter:

3 - UNCONDITIONAL SEARCH

performs an uninterrupted search of all tables in the specified library. The search is stopped immediately when you select Option 1.

2 - ASK AGAIN AFTER number TABLES

searches the specified number of tables in the specified library, and then pauses. (The search number can be modified. The default is 10.)

Do one of the following:

■ Continue the search by pressing Enter.

■ Stop the search by selecting Option 1 - Stop Search Immediately.

If any tables are found, the Table List is displayed listing those tables.

During the search, the following information is displayed at the bottom of the window:

■ Number of tables in library – the total number of tables in the specified library
Table List Screen

The Table List screen, which is shown in Figure 26, displays a list of rule tables (members) in the specified library. You can enter this screen from the entry panel or when returning from the Rule List screen.

By default, only table names are listed in the screen. However, if the default has been modified at time of installation, statistical information is displayed with the table name (as shown in Figure 26).

To scroll down the Table list, press PF08/PF20. To scroll up the Table list, press PF07/PF19.
To return to the entry panel, press PF03/PF15 (END).

**Options of the Table List Screen**

To request one of the following options, specify the option in the OPT field next to the table name, and press Enter.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(SELECT)</td>
<td>Display the list of rules in the table for any purpose, including editing and modification. Only one table can be selected at a time.</td>
</tr>
<tr>
<td>B(BROWSE)</td>
<td>Display the list of rules in a table for purposes of browsing. Only one table can be selected at a time.</td>
</tr>
<tr>
<td>D(DELETE)</td>
<td>Delete the table (member) from the library. Multiple tables can be selected.</td>
</tr>
</tbody>
</table>

**NOTE**
The INCONTROL administrator can limit specific users so they can access only the BROWSE option.

To return to the entry panel, press PF03/PF15 (END).

**Deleting Tables**

To delete tables, type D by the table names and press Enter.

The confirmation window shown in Figure 27 is displayed, in sequence, for each table selected for deletion.
Figure 27  Rule Definition Facility Delete Table Confirmation Window

<table>
<thead>
<tr>
<th>Command</th>
<th>CONTROL-M JCL VERIFY RULES - TABLES LIST</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPT NAME</td>
<td>+--------------------------+ E INIT MOD ID</td>
<td></td>
</tr>
<tr>
<td>$HASP</td>
<td>CONFIRM DELETE OPTION</td>
<td>513 38 0 IOAPROD</td>
</tr>
<tr>
<td>D COPYRULE</td>
<td>(Y/N)</td>
<td>48 10 0 IOAPROD</td>
</tr>
<tr>
<td>COPYTEST</td>
<td>+--------------------------+ 0 12 0 IOAPROD</td>
<td></td>
</tr>
<tr>
<td>COPYTST</td>
<td>01.16 2013/02/04 2013/02/13 16:46</td>
<td>52 25 0 IOATEST</td>
</tr>
<tr>
<td>CTJRSAMP</td>
<td>01.00 2013/06/05 2013/06/05 12:00</td>
<td>1 1 0 IOATEST</td>
</tr>
<tr>
<td>DEMO1</td>
<td>01.03 2013/01/30 2013/02/03 15:28</td>
<td>22 11 0 IOAPROD</td>
</tr>
<tr>
<td>ENVNAMED</td>
<td>01.10 2013/01/30 2013/02/11 09:57</td>
<td>64 9 0 IOAPROD</td>
</tr>
<tr>
<td>CTD</td>
<td>01.09 2013/01/24 2013/02/13 16:42</td>
<td>25 15 0 IOAB000</td>
</tr>
<tr>
<td>CTM</td>
<td>01.08 2013/01/24 2013/01/27 14:23</td>
<td>14 7 0 IOAB000</td>
</tr>
<tr>
<td>CTO</td>
<td>01.08 2013/01/17 2013/02/04 18:26</td>
<td>57 13 0 IOAB000</td>
</tr>
<tr>
<td>DEVICES</td>
<td>01.06 2013/01/08 2013/02/20 11:03</td>
<td>17 11 0 IOAB000</td>
</tr>
<tr>
<td>NEWRULES</td>
<td>01.01 2013/01/29 2013/01/29 14:05</td>
<td>78 63 0 IOAPROD</td>
</tr>
<tr>
<td>NEWTBLR</td>
<td>01.01 2013/02/17 2013/02/17 21:06</td>
<td>15 11 0 IOATEST</td>
</tr>
<tr>
<td>ONDD</td>
<td>01.31 2012/12/10 2013/01/20 11:12</td>
<td>26 14 0 IOATEST</td>
</tr>
<tr>
<td>ONDDS</td>
<td>01.04 2013/01/16 2013/01/20 11:11</td>
<td>16 15 0 IOAPROD</td>
</tr>
<tr>
<td>ONEXEC</td>
<td>01.08 2012/12/10 2013/01/20 12:01</td>
<td>16 14 0 IOAPROD</td>
</tr>
<tr>
<td>SHUTSYS</td>
<td>01.11 2013/01/09 2013/02/17 22:04</td>
<td>37 12 0 IOAPROD</td>
</tr>
<tr>
<td>STARTSYS</td>
<td>01.20 2012/12/05 2013/02/13 23:54</td>
<td>66 11 0 IOAPROD</td>
</tr>
<tr>
<td>STATS</td>
<td>01.07 2012/12/10 2013/01/24 17:10</td>
<td>12 13 0 IOAPROD</td>
</tr>
</tbody>
</table>

Options S SELECT B BROWSE D DELETE 14.52.55

- Type Y (Yes) in the window to delete the table.
- Type N (No) in the window to cancel the deletion request.

--- NOTE ---
If the PDSMAN product is operational at your site, $$$SPACE members cannot be deleted.

---
For each table deleted, a message is written to the IOA Log.

Rule List Screen

The Rule List screen displays the list of rules in a rule table. This screen can be entered directly from the entry panel or the Table List screen, or upon returning from the Rule Definition screen.

--- NOTE ---
If the S (Select) option was specified in the Table List screen for a table which is currently in use (“selected”) by another user, then depending on the User Profile definition, either the Rule List screen is not displayed (that is, the Table List screen remains displayed; this is the default), or the Rule List screen is displayed in Browse mode. In either case, an appropriate message is displayed.
Format of the Rule List Screen

Next to each rule name in the Rule list, certain information can be displayed. The type and format of this information depends on whether the screen is displayed in DESC format or in STAT format:

- In DESC format, the description of the rule, taken from the DESCRIPTION field of the rule definition, is displayed. This is the default.

- In STAT format, the ISPF-like statistical information relating to the rule is displayed.

To change formats, use the DESC or STAT commands. To change the order in which the rules are displayed in the Rule List screen, use the SORT R command. For more information about these commands, see Table 14 on page 73. Table 13 lists the Rule Types and their descriptions.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD</td>
<td>Indicates that the rule is triggered by a DD statement.</td>
</tr>
<tr>
<td>EXEC</td>
<td>Indicates that the rule is triggered by an EXEC statement.</td>
</tr>
<tr>
<td>INCLUDE</td>
<td>Indicates that the rule is triggered by an INCLUDE statement.</td>
</tr>
<tr>
<td>JCLLIB</td>
<td>Indicates that the rule is triggered by a JCLLIB statement.</td>
</tr>
</tbody>
</table>
Commands of the Rule List Screen

The commands in Table 14 can be entered in the COMMAND field of the Rule List screen.

Table 14 Commands of the Rule List Screen

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESC</td>
<td>The DESC command displays the rule description next to the rule name. The description is taken from the DESC field in the rule.</td>
</tr>
</tbody>
</table>
| STAT | The STAT command displays the following ISPF-like statistical information next to the rule names:  
  ■ the creation date  
  ■ the last modification date  
  ■ the version and modification numbers  
  ■ the user ID |
| SORT R | The SORT R command sorts the list of rules in the Rule List screen according to the RULE field. |

Options of the Rule List Screen

Use one of the rule options shown in Table 15 by specifying the option next to the rule name and pressing Enter.
Table 15 Options of the Rule List Screen

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S (SELECT)</td>
<td>Display the Rule Definition screen which contains details of the specific rule. If the Rule Definition screen is displayed in Browse mode, the rule definition can only be browsed; it cannot be modified. If the Rule Definition screen is not displayed in Browse mode, the rule definition can be edited and updated.</td>
</tr>
<tr>
<td>D (DELETE)</td>
<td>Delete a rule from the Rule list.</td>
</tr>
<tr>
<td>I (INSERT)</td>
<td>Insert a new rule in the list. The Rule Definition screen appears, with the same parameter values as the rule marked I.</td>
</tr>
<tr>
<td>C (COPY)</td>
<td>Copy the rule to another table (described below). Multiple rules can be selected.</td>
</tr>
</tbody>
</table>

**NOTE**
If the Rule List screen is displayed in Browse mode, options D (Delete) and I (Insert) are not available.

Copying Rules to Another Table

To copy one or more rules from the current table to another table, type C (Copy) next to the rules names in the Rule List screen and press Enter. The screen shown in Figure 29 is displayed.
Figure 29  Rule List Screen Copy Window

The window contains the fields in Table 16. Some fields contain default values which can be modified.

Table 16  Fields of the Rule List Screen Copy Window

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE</td>
<td>Name of the table into which the rule should be copied. <strong>Note:</strong> A rule can only be copied to another table. It cannot be copied to its own table (even if the rule is renamed). If the specified table does not exist, the table is created when the request is performed.</td>
</tr>
<tr>
<td>RULE</td>
<td>Name of the rule to be copied. If multiple rules are selected, the window is first displayed with the first selected rule. As each request is preformed or canceled, the next requested rule name appears.</td>
</tr>
</tbody>
</table>

- To perform a request, press **Enter**.
- To cancel a request, press **PF03/PF15 (END)** or **PF04/PF16 (RESET)**.
Rule Definition Screen

This screen is used to define, display and modify various rule parameters (such as those for a message, string, command, or event). The rule parameters determine Control-M/JCL Verify selection and action management. This screen can be entered directly from the entry panel or from the Rule List screen.

The rule parameters may take up more than one screen.

For a detailed explanation of each rule parameter, see “Parameter Descriptions” on page 87.

--- NOTE ---

Parameters marked with the symbol $^M$ can have multiple occurrences. Whenever you fill in the last occurrence of the parameter on the screen, Control-M/JCL Verify adds a new empty occurrence of the parameter that can be filled in. The only limit to the number of occurrences is the region size available for the application.

To delete a parameter on the screen, erase it (press the EOF key, or blank it out). If additional operations are required, Control-M/JCL Verify issues appropriate instructions.

The rule parameters are divided into the following basic groups:

- General parameters
- Common Selection parameters
- Event Selection parameters
- Action parameters
General Parameters

The parameters shown in Figure 31 are used to provide general information about the rule and to determine rule searching processing.

Table 17 General Parameters (part 1 of 2)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RULE</td>
<td>Name of a rule as specified in the entry panel.</td>
</tr>
<tr>
<td>ENVIRONMENT</td>
<td>Name of the environment where the rule applies.</td>
</tr>
<tr>
<td>PRIORITY</td>
<td>Specifies the priority of the rule.</td>
</tr>
</tbody>
</table>

Figure 30 Rule Definition Screen

<table>
<thead>
<tr>
<th>RULE</th>
<th>ENVIRONMENT</th>
<th>PRIORITY</th>
<th>CONTINUE SEARCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESC</td>
<td>DESC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 31 General Parameters

<table>
<thead>
<tr>
<th>RULE</th>
<th>ENVIRONMENT</th>
<th>GENERAL</th>
<th>PRIORITY</th>
<th>CONTINUE SEARCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESC</td>
<td>DESC</td>
<td></td>
<td>50</td>
<td>Y</td>
</tr>
</tbody>
</table>

Table 17 General Parameters (part 1 of 2)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RULE</td>
<td>Name of a rule as specified in the entry panel.</td>
</tr>
<tr>
<td>ENVIRONMENT</td>
<td>Name of the environment where the rule applies.</td>
</tr>
<tr>
<td>PRIORITY</td>
<td>Specifies the priority of the rule.</td>
</tr>
</tbody>
</table>
Common Selection Parameters

Common Selection parameters, shown in Figure 32, which are common to all the various ON parameters types, specify which JCL statements will trigger the validation processes to be performed by Control-M JCL Verify. Several selection blocks can be specified with Or/Not relationships. The Common Selection parameters are shown in Table 18.

![Figure 32  Common Selection Parameters](image)

Table 18  Common Selection Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBNAME</td>
<td>Name of the job (as appears in the JOB statement).</td>
</tr>
<tr>
<td>LIBRARY</td>
<td>Name of the library where the JCLs reside. (Note that this is also the MEMLIB, if invoked from a Control-M/EM interface.)</td>
</tr>
<tr>
<td>MEMBER</td>
<td>Name of the member name in the library. (Note that this is also the MEMNAME, if invoked from a Control-M/EM interface.)</td>
</tr>
<tr>
<td>SCHDLIB</td>
<td>Name of the scheduling library when invoked from a Control-M/EM interface.</td>
</tr>
<tr>
<td>SCHDTAB</td>
<td>Name of the scheduling table when invoked from a Control-M/EM interface. (Note that the name does not refer to “folder” since “table” is still the term used with Control-M for z/OS.)</td>
</tr>
<tr>
<td>O/N</td>
<td>Conjunctonal subparameter that permits linking of common selection statements. Valid values:</td>
</tr>
<tr>
<td></td>
<td>O (Or) – Only one statement need be satisfied.</td>
</tr>
<tr>
<td></td>
<td>N (Not) – The condition in the following statement need not be satisfied.</td>
</tr>
</tbody>
</table>
Event Selection Parameters

Event Selection parameters specify selection conditions that must be fulfilled before Control-M/JCL Verify will perform designated actions.

Figure 33 shows a typical screen with Event Selection parameters.

Figure 33  Event Selection Parameters

An ONM statement is comprised of the ON field and the optional parameters listed below in Table 19. At least one ON statement is required. Additional ON statements with the same type of subparameter can be specified using the Or/Not option described below. Type one of the following options in the ON field and press Enter. Specify a value for the selected option, and specify values as indicated for any additional subparameters.
### Table 19  ON statement options

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBINIT</td>
<td>Specifies that the rule is triggered when a JCL job is about to be processed. Used to initialize user variables (for example, counters).</td>
</tr>
<tr>
<td>JOBEND</td>
<td>Specifies that the rule is triggered when a JCL job is ending. Used, for example, for cross JCL validations and counter checking.</td>
</tr>
<tr>
<td>JOB</td>
<td>Specifies that the rule is triggered by a JOB statement. Used to validate the JOB statement.</td>
</tr>
<tr>
<td>INCLUDE</td>
<td>Specifies that the rule is triggered by an INCLUDE statement. Used to validate the INCLUDE statements.</td>
</tr>
<tr>
<td>JCLLIB</td>
<td>Specifies that the rule is triggered by a JCLLIB statement. Used to validate the JCLLIB statements.</td>
</tr>
<tr>
<td>EXEC</td>
<td>Specifies that the rule is triggered by an EXEC statement. Used to validate the EXEC statement.</td>
</tr>
<tr>
<td>PROC</td>
<td>Specifies that the rule is triggered by a PROC statement. Used to validate the PROC statements.</td>
</tr>
<tr>
<td>DD</td>
<td>Specifies that the rule is triggered by a DD statement. Used to validate the DD statement.</td>
</tr>
<tr>
<td>DD SYSIN</td>
<td>Specifies that the rule is triggered by a SYSIN DD statement (either DD * or DD DATA). Used to validate the SYSIN DD statements.</td>
</tr>
<tr>
<td>DD SYSOUT</td>
<td>Specifies that the rule is triggered by a SYSOUT DD statement set to a specific value. Used to validate the DD statement with SYSOUT equal to the specified value.</td>
</tr>
</tbody>
</table>

Mask characters (* and ?) can be used when specifying values for all ON selection.

Table 20 shows the O/N subparameter that is displayed for all types of ON statements.

### Table 20  ON Statement Or/Not Subparameter

<table>
<thead>
<tr>
<th>Subparameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O/N</td>
<td>Conjunctional parameter for linking ON statements.</td>
</tr>
<tr>
<td></td>
<td>Specifying O (Or), or N (Not) opens a new ON statement and links the new statement to the line containing the O/N specification.</td>
</tr>
</tbody>
</table>
Action Parameters: DO Statement

Action parameters (DO statements) specify actions to be performed by Control-M/JCL Verify. These actions are performed only after conditions specified in the Event Selection parameters have been fulfilled.

At least one DO statement must be specified in a rule. Note that the DO IF, DO ELSE and DO ENDIF statements provide Boolean “IF” logic capability, allowing for alternative actions to be performed.

A list of allowable DO options is displayed in the DO TYPES section under the Action parameters section of the Rule Definition screen. The DOOPT command can be used to alternately hide or display this list.

To delete a DO statement, place the cursor on the first letter in the DO field and press the EOF key on your keyboard or blank the field.

DO^M Options

Following are descriptions and examples of each of the DO statement options.

DO MSG – Delimit DO statements which are processed during message interception in command-response mode or for multi-line messages.

DO IF / DO ELSE / DO ENDIF – Provide Boolean “IF” logic capability allowing for alternative actions to be performed.
DO RETURN – Exit from the rule and returns to Control-M JCL Verify.

DO SET – Assign a value to a variable.

DO REXX – Call user REXX EXECs.

**Commands of the Rule Definition Screen**

Table 21 shows the commands command, that can be specified in the COMMAND field of the Rule Definition screen.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOOPT</td>
<td>The DOOPT command alternately displays or hides the list of allowable Automated Console Action parameters (DO OPTIONS).</td>
</tr>
<tr>
<td>ONOPT</td>
<td>The ONOPT command alternately displays or hides the list of allowable Event Selection parameters (ON OPTIONS).</td>
</tr>
</tbody>
</table>

**Editing Rule Definitions in the Edit Environment**

Rule Definition parameters can be edited, that is, moved, copied, deleted, or repeated, by performing IOA Line Editing commands, similar to standard ISPF line commands, from within the Control-M/JCL Verify Edit environment.
The Edit Environment in a Rule Definition screen is accessed by typing **EDIT** in the COMMAND field and pressing **Enter**.

A 2-character Line Editing command field, marked by underscores, is displayed for each line in the Rule Definition screen.

Editing commands are typed directly onto these underscores.

**Figure 34  Entering Editing Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEXTRULE (PF11/23)</td>
<td>Keep the changes made to the rule parameters and enter the rule definition of the next rule in the Rule list.</td>
</tr>
<tr>
<td>PREVRULE (PF10/22)</td>
<td>Keep the changes made to the rule parameters and enter the rule definition of the previous rule in the Rule list.</td>
</tr>
</tbody>
</table>

The line editing commands that you have typed are processed when **Enter** is pressed.

For more information and examples of editing rule definitions in the Edit environment, see Appendix C, “Editing Rule Definitions in the IOA Edit Environment,”

**Exiting the Rule Definition Screen**

The commands shown in Table 22 can be used to exit the Rule Definition screen.

**Table 22  Commands for Exiting the Rule Definition Screen (part 1 of 2)**
Exiting the Rule List Screen

Press **END (PF03/PF15)** to exit the Rule List screen. If changes have been made, the Exit Option window shown in **Figure 35** is opened.

**Figure 35 Rule List Screen Exit Option Window**

You must type **Y** (Yes) or **N** (No) in one of the two exit options, SAVE or CREATE. It is possible to change the name of the table or the name of the rule in the window. You will return either to the Table list or to the entry panel, depending on how you entered the Rule List screen.

---

**NOTE**

Changes made to rules in the table can be saved only by saving an entire table using the Exit Option window.

---

**Table 22 Commands for Exiting the Rule Definition Screen (part 2 of 2)**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>END (PF03/15)</td>
<td>Keep the changes made to the rule parameters and return to the Rule List screen. <strong>Note:</strong> The changes to the parameters are kept in memory only. They are written to the disk only after you exit the Rule List screen.</td>
</tr>
<tr>
<td>CANCEL</td>
<td>Do not keep the changes made to the rule parameters, and return to the Rule List screen.</td>
</tr>
</tbody>
</table>
If you have entered an Exit Option window, but do not want to exit the Rule List screen, press **PF04/PF16 (RESET)**. The Rule List screen is then displayed, and the table is not saved.

**Copying Rules to Another Table**

To copy one or more rules from the current table to another table, type **C** (Copy) by the rule names in the Rule List screen and press **Enter**. The window shown in Figure 36 is displayed.

**Figure 36  Window for Copying Rules to Another Table**

The fields in the window in Figure 36 are described in Table 23. Some fields contain default values which can be modified.
To perform a request, press Enter.

To cancel a request, press PF03/PF15 (END) or PF04/PF16 (RESET).

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE</td>
<td>Name of the table into which the rule will be copied. If the specified table does not exist, the table is created when the request is performed.</td>
</tr>
<tr>
<td>RULE</td>
<td>Name of the rule to be copied. If multiple rules are selected, the window is first displayed with the first selected rule. As each request is performed or canceled, the next requested rule name appears.</td>
</tr>
</tbody>
</table>
Parameter Descriptions

Rule definition parameters fall into the following basic categories:

- General Parameters – general information. Optional.
- Action Parameters – actions to be performed. Mandatory.

This section contains detailed descriptions of all rule definition parameters (in alphabetical order).
CONTINUE SEARCH: General Parameter

Determines whether to continue to search for additional rules that meet the selection criteria.

Mandatory. Valid values are shown in Table 24.

Table 24  CONTINUE SEARCH Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y (Yes)</td>
<td>After the DO actions in this rule are performed, Control-M/JCL Verify continues to search for additional rules with matching selection criteria. The search continues until a matching rule is found in which a value of N (No) is specified for CONTINUE SEARCH, or until all possible DO actions have been performed. Default.</td>
</tr>
<tr>
<td>N (No)</td>
<td>After the DO actions in this rule are performed, Control-M/JCL Verify stops searching for additional rules with matching selection criteria.</td>
</tr>
</tbody>
</table>
DESC: General Parameter

Description of the rule definition (in free text).

Optional. This parameter may consist of one or more lines of free text. After you have filled in one line, a new blank line is automatically displayed.

General Information

The DESC parameter does not operate as a selection parameter. It serves as internal documentation to aid in the identification and description of individual rules.

The first line of the rule description appears as the description of the rule in the Rule List screen, and the first 20 characters of the description are displayed in the Rule Status screen. It is therefore advisable to place the most important information at the beginning of the rule description.

Text for the DESC parameter can be specified in any language.
DO statement: Action Parameter

Actions taken when ON message or event criteria are satisfied.

Optional. Specify DO statements as follows:

- Type the action keyword (for example, MSG) in the DO field and press Enter.

- In many cases, subparameter fields are displayed. Fill in the subparameters and press Enter again.

After entering a DO statement, another DO line is automatically displayed. Multiple DO statements can be specified.

---

**NOTE**

To prevent infinite loops and performance degradation, Control-M/JCL Verify performs no more than 10000 actions in a rule before terminating the rule.

---

Table 25 shows valid DO actions. Each is discussed individually later in this chapter.

Table 25   DO Actions

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DO IF / DO ELSE / DO ENDIF</td>
<td>Provide Boolean “IF” logic capability allowing for alternative actions to be performed.</td>
</tr>
<tr>
<td>DO RETURN</td>
<td>Exit from a rule and returns to Control-M JCL Verify.</td>
</tr>
<tr>
<td>DO REXX</td>
<td>Call user exit written in REXX.</td>
</tr>
<tr>
<td>DO SET</td>
<td>Assign a value to a JCL statement or user-defined variable.</td>
</tr>
<tr>
<td>DO MSG</td>
<td>Define a message to be displayed in the validation output.</td>
</tr>
</tbody>
</table>

General Information

DO statements specify actions to be performed when the rule is triggered.

DO statements can be performed repetitively, or conditionally, or both repetitively and conditionally, for example, by using DO IF statements, which is described on page 92.

Whenever a DO statement is specified, an empty DO statement is added after it. This allows specification of an unlimited number of DO statements.
To add an empty DO statement between two existing DO statements, specify the appropriate Line Editing command in the Edit environment. For more information on the Edit environment, see Appendix C, “Editing Rule Definitions in the IOA Edit Environment.”

To delete unwanted DO statements, either delete the DO action keyword and press Enter or specify the appropriate Line Editing commands in the Edit environment. For more information on the Edit environment, see Appendix C, “Editing Rule Definitions in the IOA Edit Environment.”
DO IF / DO ELSE / DO ENDIF: Action Parameter

The DO block can optionally include IF/ELSE/ENDIF blocks. The block permits conditionally performing DO commands. DO IF, DO ELSE, and DO ENDIF statements provide Control-M/JCL Verify with Boolean “IF” logic capability. These statements permit branching based on different criteria. The examples at the end of this section demonstrate combination of these statements.

Type IF in the DO field and press Enter. The word DO is replaced by the word IF on the screen. The same will occur when specifying ELSE and ENDIF in the DO fields.

Figure 37 shows the basic format of the IF/ ELSE/ ENDIF control statements.

![Figure 37 IF/ ELSE/ ENDIF Statements Format](image)

The IF conditional expression has the following format:

**IF operand operator operand**

Valid logical operators are shown in Table 26. For numeric comparisons, use the logical operators with the "#" symbol.

Table 26 Logical Operators for IF Statement (part 1 of 2)

<table>
<thead>
<tr>
<th>Operator</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQ</td>
<td>is equal to</td>
</tr>
<tr>
<td>EQ#</td>
<td></td>
</tr>
<tr>
<td>NE</td>
<td>is not equal to</td>
</tr>
<tr>
<td>NE#</td>
<td></td>
</tr>
<tr>
<td>GT</td>
<td>is greater than</td>
</tr>
<tr>
<td>GT#</td>
<td></td>
</tr>
<tr>
<td>GE</td>
<td>is greater than or equal to</td>
</tr>
<tr>
<td>GE#</td>
<td></td>
</tr>
</tbody>
</table>
Valid Boolean operators are shown in Table 27.

Table 27  Boolean Operators for IF Statement

<table>
<thead>
<tr>
<th>Operator</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>both expressions must be true</td>
</tr>
<tr>
<td>OR</td>
<td>either expression must be true</td>
</tr>
</tbody>
</table>

Operators that end with the pound sign (#) are used for numerical comparisons, as opposed to string comparisons.

An operand can be any character string. It can also be composed of AutoEdit symbols. In such cases, it is resolved into a character string before the conditional expression is analyzed at execution time.

Whenever non-numeric comparison operators are specified, operands are compared as character strings from left to right. For example, in the expression

\[
\text{IF 91 GT 1000}
\]

91 is greater than 1000 (because 9 is greater than 1).

An operand cannot be resolved into nulls (as in CLISTs). If it is possible that an operand will resolve into nulls, place a character before the first and second operands.

In the following example, the character B is placed before the two operands:

\[
\text{IF B%%A GT B%%C}
\]

An IF expression must be terminated with an ENDIF statement. The ELSE statement is optional.

**General Information**

When the IF expression is true, commands between the IF expression and its ELSE statement, or its matching ENDIF statement when no ELSE statement is present, are resolved by Control-M/JCL Verify.
When the IF expression is not true, and an ELSE statement exists, only the statements after the ELSE statement are executed. If the ELSE clause does not exist, Control-M/JCL Verify stops processing the IF expression and continues executing the rule.

IF expressions can be nested, providing that the above rules are observed. Up to 100 nested IF statements are permitted.

When the rule is saved, logical checks are performed to verify the validity of the rule statements. If an error is detected, an error message is issued upon exiting the rule, and the rule cannot be saved until the syntax is corrected.

Example

Example 1

In this example, if the account field in the job card is empty, we update it to “0000”. If the class is not U or D, we issue an error message. The jobs that are tested are jobs that begin with PAY* and are submitted from the PRD.PAYACCOUNT.DAILY.JOBS library from members that begin with DPAY*.

Figure 38 DO IF example 1

```
+-----------------------------------------------------------------------------+
| RULE DAILYPAY   ENVIRONMENT PROD       PRIORITY 3 CONTINUE SEARCH Y    |
| DESC        For accounting jobs, enforce account information and            |
| DESC        class D and U                                                   |
|=============================================================================|
| JOBNAME PAY*____            _                                               |
| LIBRARY  PRD.PAYACCOUNT.DAILY.JOBS__________________  MEMBER  DPAY*___     |
| SCHDLIB                                              SCHDTAB          O/N   |
|=============================================================================|
| ON JOB           CLASS         PRTY          USER                           |
| ACCOUNT                                                               O/N   |
|=============================================================================|
| IF %%JOB_ACT EQ ''                                                          |
| DO MSG  TYPE E Account field is empty.Automatically filled with “0000”.     |
| DO SET  %%JOB_ACT = '0000'                                                  |
| ENDIF                                                                       |
| IF %%$JOBCLS NE 'D'                                                         |
| IF %%$JOBCLS NE 'U'                                                         |
| DO MSG  TYPE E JOBS of PAYABLE must run with class D or U.                  |
| ENDIF                                                                       |
| ENDIF                                                                       |
|=============================================================================|
```
DO MSG: Action Parameter

Adds messages to the verifier output.

The valid format is

```
MSG TYPE type text
```

where

- **type** can be one of the following: I (Information), W (warning), or E (Error)

- **text** is the message that is added to the output using a JCL Verify message ID that ends with I/W/E based on the type above.

Optional. Type **MSG** in the **DO** field, and press **Enter**. The **TYPE** field is displayed. Enter one of the valid values: I, W, or E. The maximum length of MSG is

- 107 characters (when MSG type is ERROR)
- 105 characters (when MSG type is WARNING)
- 101 characters (when MSG type is INFORMATION)

**Example**

**Example 1**

In this example, if the class is not U or D, we issue an error message. The jobs that are tested are jobs that begin with PAY* and are submitted from the PRD.PAY.ACCOUNT.DAILY.JOBS library from members that begin with DPAY*.

**Figure 39  DO MSG Parameter Example**

```plaintext
+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+
| RULE   DAILYPAY ENVIRONMENT PROD | PRIORITY 3 CONTINUE SEARCH Y   | DESC          Warns that accounting jobs must run with class D and U. | |
+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+
| JOBNAME PAY*                     | LIBRARY PRD.PAY.ACCOUNT.DAILY.JOBS____ MEMBER DPAY*___ | SCHDLIB                                               | SCHDTAB                | D/N                             |
+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+
| ON JOB CLASS PRTY USER ACCOUNT   | CLASS PRTY USER ACCOUNT         | D/N                             |
+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+
| IF %$JOBCLS NE 'D'              | IF %$JOBCLS NE 'U'              | DO MSG TYPE E Accounting jobs must run with class D or U. | ENDIF
+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+---------------------------------+
```
DO RETURN: Action Parameter

Exits from the rule and returns to Control-M/JCL Verify.

The valid format is

```
RETURN RC 0000 RS X' 0000 ' 
```

where RC refers to the highest return code attained up until the current point in the procedure, and RS refers to the highest reason code (expressed in HEX code) attained up until the current point in the procedure.

Optional. Type `RETURN` in the DO field, and press `Enter`.

Example

Example 1

In this example, we ensure that all the datasets of the Payment Department jobs have a HLQ of PAYDY, all the datasets are on 3390 disks, and the space is always allocated in TRKS only. After the test, the process exits from the rule and returns to Control-M/JCL Verify.

Figure 40  DO RETURN example

```plaintext
RULE DAILYPAY ENVIRONMENT PROD PRIORITY 3 CONTINUE SEARCH Y
DESC Checks files of payable jobs

+-----------------------------------------------+
<p>| JOBNAME PAY*____            _               |
| LIBRARY  PRD.PAY.ACCOUNT.DAILY.JOBS___________ |
| SCHDLIB | SCHDTAB | O/N |
|-----------------------------------------------------------------------------|
| ON DD       *        DSN *                                                  |
|    DISP                     UNIT            VOL                             |</p>
<table>
<thead>
<tr>
<th>SMS CLASS DATA           STORAGE          MANAGEMENT               O/N</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF %%$SUBSTR %%$DDSN 1 6 NE 'PAYDY.'</td>
</tr>
<tr>
<td>DO MSG  TYPE E Invalid HLQ for payable system files.</td>
</tr>
<tr>
<td>ENDIF</td>
</tr>
<tr>
<td>IF %%$DDUNIT NE '3390'</td>
</tr>
<tr>
<td>DO MSG  TYPE I UNIT should be 3390 for all payable files.</td>
</tr>
<tr>
<td>ENDIF</td>
</tr>
<tr>
<td>IF %%$GETWORD %%$SPCUNIT 1 ' ' NE TRKS</td>
</tr>
<tr>
<td>DO MSG  TYPE I SPACE should be allocated in TRKS only.</td>
</tr>
<tr>
<td>ENDIF</td>
</tr>
<tr>
<td>RETURN   RC 0000 RS X' 0000 '</td>
</tr>
</tbody>
</table>
+-----------------------------------------------+ 
```
DO REXX: Action Parameter

Calls REXX EXECs.

The valid format is

\[
\text{DO REXX MEM exec LIB library PARM parameters}
\]

where

- \textit{exec} (optional) is the member name of the REXX EXEC

- \textit{library} is the library name where the EXEC resides. If LIB is not specified then the EXEC is searched in DD CARD SYSEXEC.

- \textit{parameters} (optional) are the parameters passed to the EXEC

Optional. Type REXX in the DO field, and press Enter.

Example

Example 1

In this example, we ensure that all the datasets of the Payment Department jobs have a HLQ of PAYDY. After the test, the process calls the TEST1 user exit, with the arguments, ARG1, ARG2, and ARG3.

\begin{verbatim}
Figure 41 DO REXX example
+-----------------------------------------------------------------------------+
| RULE DAILYPAY   ENVIRONMENT PROD       PRIORITY 3 CONTINUE SEARCH Y    |
| DESC       Checks files of payable jobs                                     |
|=============================================================================|
| JOBNAME PAY*____            _                                               |
| LIBRARY  PRD.PAY.ACCOUNT.DAILY.JOBS__________________  MEMBER  DPAY*___     |
| SCHDLIB                                              SCHDTAB          O/N   |
|=============================================================================|
| ON DD       *        DSN *                                                  |
|    DISP                     UNIT            VOL                             |
|    SMS CLASS DATA           STORAGE          MANAGEMENT               O/N   |
|=============================================================================|
| IF %%$SUBSTR %%$DDDSN 1 6 NE 'PAYDY.'                                       |
| DO MSG  TYPE E Invalid HLQ for payable system files.                        |
| ENDIF                                                                       |
| DO REXX MEM TEST1 LIB SYS2.CLIST                                            |
| PARM ARG1 ARG2 ARG3                                                         |
+-----------------------------------------------------------------------------+
DO SET: Action Parameter

Assigns a value to a variable.

The following format is used:

\[
\text{SET } var = \text{operand}
\]

where \( \text{operand} \) is a variable, constant, or built-in function.

Optional. Type \text{SET} in the \text{DO} field and press \text{Enter}.

While the user cannot change SYSTEM variables using DO SET, the built-in functions shown in Table 28 can be used to assign values to the other variable types.

Table 28  Built-in functions (part 1 of 8)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| GETWORD  | Gets the value of a JCL parameter from a predefined variable. Valid format: \%

\%

var = \%

$\text{GETWORD } %\%\text{var num}

where

\( \text{var} \) - variable name (can be a user defined variable)

\( \text{num} \) - the sequential subparameter (If there are more than one subparameters, for example, DSN and DCB.) Maximum value is 128.

| LENGTH   | Returns the length of string. Valid format: \%

\%

var = \%

$\text{LENGTH } \text{argument}

where

\( \text{argument} \) - a variable name or a built-in function
### Table 28  Built-in functions (part 2 of 8)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| POS      | Obtains the position of one string in another  
             Valid format:  
             \[
             \%\%var = \%\%$POS \%\%var \ operand \ start
             \]  
             where  
             \*operand\* - a variable name or a constant  
             \*start\* - search starting position in the \%\%var. Maximum value is 128. |
| SUBSTR   | Returns the substring of string that begins at the \*n\*th character and is of defined length, padded with pads if necessary.  
             Valid format:  
             \[
             \%\%var = \%\%$SUBSTR \%\%var \ start \ length
             \]  
             where  
             \*start\* - search starting position in the \%\%var  
             \*length\* - length of the substring (default- to the end of the \%\%var) |
| TIME     | Returns the current time in \*HHMSS\* format.  
             Valid format:  
             \[
             \%\%var = \%\%$TIME
             \] |
| UPPER    | Convert a string into upper case characters.  
             Valid format:  
             \[
             \%\%var = \%\%$UPPER \%\%var
             \] |
| LOWER    | Convert a string into lower case characters.  
             Valid format:  
             \[
             \%\%var = \%\%$LOWER \%\%var
             \] |
### Table 28  Built-in functions (part 3 of 8)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| **NUM**  | Return the numeric value of a variable. Used for converting values that appear in such variables as: REGION, MEMLIMIT, BLKSIZE. (For example, 10M can be converted to 10000000). Valid format: 

```
%%%var = %%%$NUM %%%var
```

| **CALCDATE** | Performs date calculations based on a given date. Valid format: 

```
%%%var = %%%$CALCDATE date ±quantity
```

where

- *date* - must be in the format *yymmdd*
- *quantity* - is a number, or numeric AutoEdit expression, of days to add to or subtract

| **ISNUM** | Indicates whether a specified string is numeric. In order to be considered numeric, the specified string must contain only numeric characters from 0 through 9, because a numeric string cannot contain letters, blanks, or special characters. Possible values returned by this function are

- **YES** — the specified string is numeric
- **NO** — the specified string is not numeric

Valid format: 

```
%%%var = %%%$ISNUM string
```
The \%\%PARSE function is a powerful tool that offers extensive string manipulation capabilities. This function, which is similar to the REXX PARSE command in the TSO/E environment, can be used to analyze and extract information from various variables strings.

The \%\%PARSE function parses a specified string (that is, it splits the specified string into substrings) according to a specified template. A template consists of variables and "patterns" that determine the parsing process.

Note that variable \%\%PARSRC contains how the function ended. It is recommended that whenever using \%\%PARSE to always check the variable \%\%PARSRC.

Valid format:

\%\%var = \%\%PARSE string template

where

\textit{string} - is the AutoEdit variable that contains the string to be parsed

\textit{template} - is the AutoEdit variable or constant that contains the template

Example 1

DO SET \%\%S=THIS IS A SAMPLE STRING
DO SET \%\%T=A1 A2 A3 A4 A5
DO SET \%\%PARSE \%\%S \%\%T

The \%\%PARSE function assigns substrings of the specified string to the specified variables according to the specified template.

The DO SET statements in the above example provide the same result as the following DO SET statements:

DO SET \%\%A1=THIS
DO SET \%\%A2=IS
DO SET \%\%A3=A
DO SET \%\%A4=SAMPLE
DO SET \%\%A5=STRING
Example 2
This example is similar to Example 1, except that a comma is used as the delimiter between words.

If the string includes commas, instead of blanks, as in the following:

```
DO SET %%S=THIS,IS,A,SAMPLE,STRING
```

use the following pattern to achieve the same results as in Example 1 above:

```
DO SET %%T=A1 , A2 , A3 , A4 , A5
DO SET %%$PARSE %%S %%T
```

For more details about the %%$PARSE function, see Appendix D, “The %%$PARSE function.”

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARSE</td>
<td>Example 2</td>
</tr>
<tr>
<td>(continued)</td>
<td>This example is similar to Example 1, except that a comma is used as the delimiter between words.</td>
</tr>
<tr>
<td></td>
<td>If the string includes commas, instead of blanks, as in the following:</td>
</tr>
<tr>
<td></td>
<td>DO SET %%S=THIS,IS,A,SAMPLE,STRING</td>
</tr>
<tr>
<td></td>
<td>use the following pattern to achieve the same results as in Example 1 above:</td>
</tr>
<tr>
<td></td>
<td>DO SET %%T=A1 , A2 , A3 , A4 , A5</td>
</tr>
<tr>
<td></td>
<td>DO SET %%$PARSE %%S %%T</td>
</tr>
<tr>
<td></td>
<td>For more details about the %%$PARSE function, see Appendix D, “The %%$PARSE function.”</td>
</tr>
<tr>
<td>PARSRC</td>
<td>The return code resulting from a %%$PARSE function that indicates whether the parsed string matched all string patterns in the template.</td>
</tr>
<tr>
<td></td>
<td>Valid values returned by this function are</td>
</tr>
<tr>
<td></td>
<td>■ 0 — The parsed string fully matched the string patterns in the template.</td>
</tr>
<tr>
<td></td>
<td>■ 4 — At least one string pattern in the template was not matched.</td>
</tr>
<tr>
<td>NULL</td>
<td>A null variable.</td>
</tr>
<tr>
<td></td>
<td>Valid format:</td>
</tr>
<tr>
<td></td>
<td>%%%var = %%%NULL</td>
</tr>
<tr>
<td>BLANK</td>
<td>Resolves to one blank.</td>
</tr>
<tr>
<td></td>
<td>Valid format:</td>
</tr>
<tr>
<td></td>
<td>%%%var = %%%BLANK</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BLANK(n)</td>
<td>Resolves to (n) blanks, where (n) is a number from 1 through 99.</td>
</tr>
<tr>
<td></td>
<td>Valid format: (%%var = %%$BLANK(n))</td>
</tr>
<tr>
<td>WORDS</td>
<td>The number of words in the (varname) variable. A comma or a blank can serve as a delimiter within the variable.</td>
</tr>
<tr>
<td></td>
<td>Valid format: (%%var = %%$WORDS varname)</td>
</tr>
<tr>
<td>%%PLUS</td>
<td>Arithmetic operator that adds two operands.</td>
</tr>
<tr>
<td></td>
<td>Valid format: (\text{DO SET } %%var = %%var %%$PLUS n)</td>
</tr>
<tr>
<td></td>
<td>Example: (\text{DO SET } %%I = %%I %%$PLUS 1)</td>
</tr>
<tr>
<td>%%MINUS</td>
<td>Arithmetic operator that subtracts one operand from another operand.</td>
</tr>
<tr>
<td></td>
<td>Valid format: (\text{DO SET } %%var = %%var %%$MINUS n)</td>
</tr>
<tr>
<td></td>
<td>Example: (\text{DO SET } %%I = %%I %%$MINUS 1)</td>
</tr>
<tr>
<td></td>
<td>Note: Negative values are not supported, therefore the results are always absolute. For example:</td>
</tr>
<tr>
<td></td>
<td>(1 %%$MINUS 24 \Rightarrow 23) (not -23)</td>
</tr>
</tbody>
</table>
Table 28  Built-in functions (part 7 of 8)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>%%$TIMES</code></td>
<td>Arithmetic operator that multiplies one operand by another operand.</td>
</tr>
<tr>
<td></td>
<td>Valid format:</td>
</tr>
<tr>
<td></td>
<td>DO SET <code>%%var = %%var %%$TIMES n</code></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>DO SET <code>%%I = %%I %%$TIMES 1</code></td>
</tr>
<tr>
<td><code>%%$DIV</code></td>
<td>Arithmetic operator that divides one operand by another operand.</td>
</tr>
<tr>
<td></td>
<td>Valid format:</td>
</tr>
<tr>
<td></td>
<td>DO SET <code>%%var = %%var %%$DIV n</code></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>DO SET <code>%%I = %%I %%$DIV 1</code></td>
</tr>
<tr>
<td><code>%%$D2X</code></td>
<td>Decimal value-to-hexadecimal string convertor. The maximum number that can</td>
</tr>
<tr>
<td></td>
<td>be converted with this variable is 2147483647, or $2^{31}$-1.</td>
</tr>
<tr>
<td></td>
<td>Valid format:</td>
</tr>
<tr>
<td></td>
<td><code>%%var = %%$D2X num</code></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>To convert decimal 4095 to the equivalent hexadecimal string (FFF), specify</td>
</tr>
<tr>
<td></td>
<td><code>%%$D2X 4095</code>.</td>
</tr>
</tbody>
</table>
Table 28  Built-in functions  (part 8 of 8)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| %%$X2D   | Hexadecimal string-to-decimal value convertor. The result of the conversion is always positive. The highest number that can be converted using this variable is 7FFFFFFF.  

Valid format:

```
%%var = %%$X2D string
```

Example:

To convert the hexadecimal string “FFF” to the equivalent decimal value (4095), specify %%$X2D FFF.

| %%$X2C   | Hexadecimal string-to-character convertor.  

Valid format:

```
%%var = %%$X2C string
```

Example:

To convert the hexadecimal string “2A4F” to text format, specify %%$X2C 2A4F.

General Information

The DO SET parameter permits setting of values to variables (or symbols). These variables can then be referenced in other DO statements in the rule definition during rule activation.

Examples

Example 1

In this example, if the account field in the job card is empty, we update it to “0000”. The jobs that are tested are jobs that begin with PAY* and are submitted from PRD.PAYACCOUNT.DAILY.JOBS library from members that begin with DPAY*.
### Figure 42  DO SET Example

```
<table>
<thead>
<tr>
<th>RULE</th>
<th>DAILYPAY</th>
<th>ENVIRONMENT PROD</th>
<th>PRIORITY 3 CONTINUE SEARCH Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESC</td>
<td>Enforces account information</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOBNAME</td>
<td>PAY*</td>
<td>LIBRARY</td>
<td>SCHDLIB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROD.PAY.ACCOUNT.DAILY.JOBS</td>
<td>SCHDLIB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MEMBER</td>
<td>O/N</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DPAY*</td>
<td>O/N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON JOB</td>
<td>CLASS</td>
<td>PRTY</td>
<td>USER</td>
</tr>
<tr>
<td>ACCOUNT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IF %%JOB_ACT EQ ''</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DO MSG TYPE E Account field is empty. Automatically filled with “0000”.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DO SET %%JOB_ACT = '0000'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENDIF</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
ENVIRONMENT: General Parameter

Name of the environment where the rule applies. Using the ENVIRONMENT parameter, the user can limit the site standard verification to use only the rules in which the particular environment name is specified. The environment can be specified with up to 8 alphanumeric characters (A-Z, 0-9, upper case only).

General Information

The ENVIRONMENT parameter is mandatory.

To create a general rule that applies to all environments, enter the keyword GENERAL in the ENVIRONMENT field.

When Control-M/JCL Verify is run to verify the site standards, if an environment is specified with the invocation parameters, both the rules that have been defined for the particular environment and the rules that have been defined with the GENERAL keyword are applied in the verification. If an environment is not specified, only the rules that have been defined with the GENERAL keyword are applied in the verification.

Examples

Example 1

In this example, we ensure that in the PROD environment, all the datasets of the Payment Department jobs have a HLQ of PAYDY, all the datasets are on 3390 disks, and the space is always allocated in TRKS only.
Figure 43  ENVIRONMENT example

<table>
<thead>
<tr>
<th>RULE</th>
<th>DAILYPAY</th>
<th>ENVIRONMENT PROD</th>
<th>PRIORITY 3 CONTINUE SEARCH Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESC</td>
<td>Checks files of payable jobs</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>JOBNAME</th>
<th>PAY*____</th>
<th>_</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIBRARY</td>
<td>PRD.PAY.ACCOUNT.DAILY.JOBS, MEMBER DPA**___</td>
<td></td>
</tr>
<tr>
<td>SCHDLIB</td>
<td>SCHDTAB</td>
<td>O/N</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ON DD</th>
<th>*</th>
<th>DSN *</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISP</td>
<td>UNIT</td>
<td>VDL</td>
</tr>
<tr>
<td>SMS</td>
<td>CLASS DATA</td>
<td>STORAGE MANAGEMENT</td>
</tr>
</tbody>
</table>

IF %%$SUBSTR %%$DDSN 1 6 NE 'PAYDY.'
DO MSG TYPE E Invalid HLQ for payable system files.
ENDIF

IF %%$DDUNIT NE '3390'
DO MSG TYPE I UNIT should be 3390 for all payable files.
ENDIF

IF %%$GETWORD %$SPCUNIT 1 ' ' NE TRKS
DO MSG TYPE I SPACE should be allocated in TRKS only.
ENDIF
RETURN RC 0000 RS X' 0000 '
JOBNAME: Common Selection Parameter

Name of the job (as appears in the JOB statement).

General Information

This optional selection parameter is common to all the various ON parameter types.

Examples

Example 1

In this example, we ensure that all the jobnames that begin with PAY have a HLQ of PAYDY, all the datasets are on 3390 disks, and the space is always allocated in TRKS only.

Figure 44  JOBNAME example

```
+-----------------------------------------------------------------------------+
| RULE DAILYPAY ENVIRONMENT PROD       PRIORITY 3 CONTINUE SEARCH Y    |
| DESC Check files of payable jobs                                           |
+=============================================================================|
| JOBNAME PAY*____            _                                               |
| LIBRARY  PRD.PAY.ACCOUNT.DAILY.JOBS__________________  MEMBER  DPAY*___     |
| SCHDLIB                                              SCHDTAB          O/N   |
+=============================================================================|
| ON DD       *        DSN *                                                  |
|    DISP                     UNIT            VOL                             |
|    SMS CLASS DATA           STORAGE          MANAGEMENT               O/N   |
+=============================================================================|
| IF %%$SUBSTR %%$DDDSN 1 6 NE 'PAYDY.'                                       |
| DO MSG  TYPE E Invalid HLQ for payable system files.                        |
| ENDIF                                                                       |
| IF %%$DDUNIT NE '3390'                                                      |
| DO MSG  TYPE I UNIT should be 3390 for all payable files.                   |
| ENDIF                                                                       |
| IF %%$GETWORD %%$SPCUNIT 1 ' '  NE TRKS                                      |
| DO MSG  TYPE I SPACE should be allocated in TRKS only.                      |
| ENDIF                                                                       |
| RETURN   RC 0000 RS X' 0000 '                                               |
+=============================================================================|
```
LIBRARY: Common Selection Parameter

Name of the library where the JCLs reside. (Note that this is also the MEMLIB, if invoked from a Control-M/EM interface.)

General Information

This optional selection parameter is common to all the various ON parameter types.

Examples

Example 1

In this example, we ensure that all the jobs in the PRD.PAY.ACCOUNT.DAILY.JOBS library, with jobnames which begin with PAY, have a HLQ of PAYDY, all the datasets are on 3390 disks, and the space is always allocated in TRKS only.

Figure 45 LIBRARY example

```
+-----------------------------------------------------------------------------+
<p>| RULE DAILYPAY ENVIRONMENT PROD PRIORITY 3 CONTINUE SEARCH Y                |
| DESC Checks files of payable jobs                                          |
|=============================================================================|
| JOBNAME PAY<em>______ LIBRARY PRD.PAY.ACCOUNT.DAILY.JOBS__ MEMBER DPAY</em>____ |
| SCHDLIB SCHDTAB O/N                                                       |
|=============================================================================|
| ON DD * DSN *                                                             |</p>
<table>
<thead>
<tr>
<th>DISP SMS CLASS DATA UNIT STORAGE MANAGEMENT O/N</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF %%$SUBSTR %%$DDDSN 1 6 NE 'PAYDY.'</td>
</tr>
<tr>
<td>DO MSG TYPE E Invalid HLQ for payable system files.</td>
</tr>
<tr>
<td>ENDIF</td>
</tr>
<tr>
<td>IF %%$DDUNIT NE '3390'</td>
</tr>
<tr>
<td>DO MSG TYPE I UNIT should be 3390 for all payable files.</td>
</tr>
<tr>
<td>ENDIF</td>
</tr>
<tr>
<td>IF %%GETWORD %%$SPCUNIT 1 ' ' NE TRKS</td>
</tr>
<tr>
<td>DO MSG TYPE I SPACE should be allocated in TRKS only.</td>
</tr>
<tr>
<td>ENDIF</td>
</tr>
<tr>
<td>RETURN RC 0000 RS X' 0000 '</td>
</tr>
</tbody>
</table>
+-----------------------------------------------------------------------------|
```
MEMBER: Common Selection Parameter

Name of the member name in the library. (Note that this is also the MEMNAME, if invoked from a Control-M/EM interface.)

General Information

This optional selection parameter is common to all the various ON parameter types.

Examples

Example 1

In this example, we ensure that all the jobs in the PRD.PAY.ACCOUNT.DAILY.JOBS library, in members beginning with DPAY, and with jobnames which begin with PAY, have a HLQ of PAYDY, all the datasets are on 3390 disks, and the space is always allocated in TRKS only.

Figure 46 MEMBER example

```
RULE DAILYPAY ENVIRONMENT PROD PRIORITY 3 CONTINUE SEARCH Y
DESC Checks files of payable jobs

JOBNAME PAY*
LIBRARY PRD.PAY.ACCOUNT.DAILY.JOBS MEMBER DPAY*
SCHDLIB SCHDTAB O/N

ON DD * DSN *
DISP CLASS DATA UNIT VOL
SMS STORAGE MANAGEMENT O/N

IF %%%$SUBSTR %%%$DDDSN 1 6 NE 'PAYDY.'
DO MSG TYPE E Invalid HLQ for payable system files.
ENDIF
IF %%%$DDUNIT NE '3390'
DO MSG TYPE I UNIT should be 3390 for all payable files.
ENDIF
IF %%%GETWORD %%%$SPCUNIT 1 ' ' NE TRKS
DO MSG TYPE I SPACE should be allocated in TRKS only.
ENDIF
RETURN RC 0000 RS X' 0000 '
```
ON DD: Event Selection Parameter

Specifies that the rule is triggered by a DD statement. Used to validate the DD statement.

Optional. Type DD in the ON field and press Enter. The subparameters (selection Fields) shown in Table 29 are displayed.

Table 29 ON DD Subparameters

<table>
<thead>
<tr>
<th>Subparameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDNAME</td>
<td>The DD card name</td>
</tr>
<tr>
<td>DSN</td>
<td>The dataset referred to by the card</td>
</tr>
<tr>
<td>DISP</td>
<td>The specified dataset disposition. The valid values are:</td>
</tr>
<tr>
<td></td>
<td>- For DISP field 1 one of the following: NEW, MOD, OLD, SHR</td>
</tr>
<tr>
<td></td>
<td>- For DISP field 2 one of the following: DELETE, KEEP, CATLG, UNCATLG</td>
</tr>
<tr>
<td></td>
<td>- For DISP field 3 one of the following: DELETE, KEEP, CATLG, UNCATLG</td>
</tr>
<tr>
<td>UNIT</td>
<td>The specified UNIT</td>
</tr>
<tr>
<td>VOL</td>
<td>The specified Volume Serial.</td>
</tr>
<tr>
<td></td>
<td>There are four VOLSERS for each ON statement and the relation between these VOLSERS is OR logic, so that if at least one of the volsers satisfy the criteria, the DO statement will be activated.</td>
</tr>
<tr>
<td></td>
<td>If you use N (Not) at the END of the ON statement, it affects the next four VOLSERS.</td>
</tr>
<tr>
<td>SMS CLASS</td>
<td>One of the following classes:</td>
</tr>
<tr>
<td></td>
<td>- DATA</td>
</tr>
<tr>
<td></td>
<td>- STORAGE</td>
</tr>
<tr>
<td></td>
<td>- MANAGEMENT</td>
</tr>
<tr>
<td>O/N</td>
<td>Conjunctonal subparameter that permits linking of ON statements. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>- O  (Or) - Only one statement need be satisfied.</td>
</tr>
<tr>
<td></td>
<td>- N  (Not) - The condition in the following statement need not be satisfied.</td>
</tr>
</tbody>
</table>

The JCL statement (DD card) variables that can be used with the ON DD statement are shown in Table 30.
### Examples

**Example 1**

In this example, we ensure that all the jobs in the PRD.PAY.ACCOUNT.DAILY.JOBS library, in members beginning with DPAY, and with jobnames which begin with PAY, have a HLQ of PAYDY, all the datasets are on 3390 disks, and the space is always allocated in TRKS only.

#### Table 30 ON DD variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>$DDNAME</code></td>
<td>Data definition name</td>
</tr>
<tr>
<td><code>$DDSN</code></td>
<td>Dataset name</td>
</tr>
<tr>
<td><code>$DDSN#</code></td>
<td>Number of qualifiers in dataset name</td>
</tr>
<tr>
<td><code>$DDSN$n</code></td>
<td>qualifiers $n$ in dataset name</td>
</tr>
<tr>
<td><code>$DDTCLS</code></td>
<td>Dataset data class value</td>
</tr>
<tr>
<td><code>$DMNGCLS</code></td>
<td>Dataset management class value</td>
</tr>
<tr>
<td><code>$STORCLS</code></td>
<td>Dataset storage class value</td>
</tr>
<tr>
<td><code>$DDISP1</code></td>
<td>Dataset first disposition value</td>
</tr>
<tr>
<td><code>$DDISP2</code></td>
<td>Dataset second disposition value</td>
</tr>
<tr>
<td><code>$DDISP3</code></td>
<td>Dataset third disposition value</td>
</tr>
<tr>
<td><code>$DDTYPE</code></td>
<td>DD card type. If it is a DUMMY DD card, its value is D.</td>
</tr>
<tr>
<td><code>$DDUNIT</code></td>
<td>Dataset UNIT value</td>
</tr>
<tr>
<td><code>$DDVOL$n</code></td>
<td>Dataset VOLUME value – $n$ is the seq.</td>
</tr>
<tr>
<td><code>$DDVOL#$</code></td>
<td>number of Dataset VOLUME value</td>
</tr>
<tr>
<td><code>$SPCUNIT</code></td>
<td>Dataset space unit</td>
</tr>
<tr>
<td><code>$SPCPRI</code></td>
<td>Dataset space primary value</td>
</tr>
<tr>
<td><code>$SPCSEC</code></td>
<td>Dataset space secondary value</td>
</tr>
<tr>
<td><code>$SPCDIR</code></td>
<td>Dataset space directory blocks value</td>
</tr>
<tr>
<td><code>$DSORG</code></td>
<td>Dataset DSORG value</td>
</tr>
<tr>
<td><code>$RECFM</code></td>
<td>Dataset RECFM value</td>
</tr>
<tr>
<td><code>$LRECL</code></td>
<td>Dataset LRECL value</td>
</tr>
<tr>
<td><code>$BLKSZ</code></td>
<td>Dataset BLKSZ value</td>
</tr>
<tr>
<td><code>$DDCONCAT</code></td>
<td>Contains Y if the DD statement is part of a concatenation of DDs.</td>
</tr>
<tr>
<td><code>$DDCONNUM</code></td>
<td>When <code>$DDCONCAT</code> is set to Y, this field contains the DD sequence number in the concatenation</td>
</tr>
</tbody>
</table>
Figure 47  ON DD example

```
<p>| RULE  DAILYPAY  ENVIRONMENT PROD  PRIORITY 3 CONTINUE SEARCH Y |
| DESC check files of payable jobs                                       |</p>
<table>
<thead>
<tr>
<th>DESC</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBNAME PAY*____   _</td>
</tr>
<tr>
<td>LIBRARY  PRD.PAY.ACCOUNT.DAILY.JOBS__________________  MEMBER  DPAY*___</td>
</tr>
<tr>
<td>SCHDLIB  SCHDTAB  O/N</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ON DD  *       DSN *</td>
</tr>
<tr>
<td>DISP UNIT VDL</td>
</tr>
<tr>
<td>SMS CLASS DATA STORAGE MANAGEMENT O/N</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IF %%$SUBSTR %%$DDSN 1 6 NE 'PAYDY.'</td>
</tr>
<tr>
<td>DO MSG  TYPE E Invalid HLQ for payable system files.</td>
</tr>
<tr>
<td>ENDIF</td>
</tr>
<tr>
<td>IF %%$DDUNIT NE '3390'</td>
</tr>
<tr>
<td>DO MSG  TYPE I UNIT should be 3390 for all payable files.</td>
</tr>
<tr>
<td>ENDIF</td>
</tr>
<tr>
<td>IF %%GETWORD %%$SPCUNIT 1 ' ' NE TRKS</td>
</tr>
<tr>
<td>DO MSG  TYPE I SPACE should be allocated in TRKS only.</td>
</tr>
<tr>
<td>ENDIF</td>
</tr>
<tr>
<td>RETURN RC 0000 RS X' 0000 '</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
```
ON DDSYSIN: Event Selection Parameter

Specifies that the rule is triggered by a SYSIN DD statement (either DD * or DD DATA). Used to validate the SYSIN DD statements.

Optional. Type DDSYSIN in the ON field and press Enter.

The following selection fields (subparameters) are displayed:

- **ddname** – the SYSIN DD name
- **DLM** - the end of SYSIN delimiter string

The following JCL statement (JOB card) variable can be used with the ON DDSYSIN statement:

```
%%$INTYPE - the SYSIN type (* or D for "DATA")
```

**Example**

In this example a message is displayed for each SYSIN DD statement.

**Figure 48  ON DDSYSIN example**

<table>
<thead>
<tr>
<th>RULE TH006</th>
<th>ENVIRONMENT THOM</th>
<th>PRIORITY 50</th>
<th>CONTINUE SEARCH Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOBNAME</td>
<td>BMC459*</td>
<td>MEMBER</td>
<td>D/N</td>
</tr>
<tr>
<td>LIBRARY</td>
<td>SCHDLIB</td>
<td>SCHDTAB</td>
<td>D/N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON DDSYSIN</td>
<td>*</td>
<td>DLM *</td>
<td>D/N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DO MSG</td>
<td>TYPE I INTYPE = %$INTYPE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DO</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ON DDSYSOUT: Event Selection Parameter

Specifies that the rule is triggered by a SYSOUT DD statement set to a specific value. Used to validate the DD statement with SYSOUT equal to the specified value.

Optional. Type **DDSYSOUT** in the **ON** field and press **Enter**.

The following selection fields (subparameters) are displayed:

- **ddname** – the SYSOUT DD name
- **CLASS** – The SYSOUT class

The following JCL statement (JOB card) variable can be used with the **ON DDSYSOUT** statement:

```
%%$DDSYSOUT – the SYSOUT information
```

**Example**

In this example a message with the SYSOUT information is displayed for SYSOUT DD statements labelled P and SYSOUT class P.

**Figure 49  ON DDSYSOUT example**

```
+---------------------------------------------------------------+
| RULE DDSYSOUT ENVIRONMENT GENERAL PRIORITY 50 CONTINUE SEARCH Y |
| DESC DDSYSOUT                                                   |
+-----------------------------------------------------------------+
| JOBNAME *                                                      |
| LIBRARY                                                        |
| SCHDLIB                                                        |
| SCHDTAB O/N                                                    |
+-----------------------------------------------------------------+
| ON DDSYSOUT P CLASS P                                          |
+-----------------------------------------------------------------+
| DO MSG TYPE I OUTTYPE = %%$DDSYSOUT                            |
| DO                                                             |
+---------------------------------------------------------------+
```
ON EXEC: Event Selection Parameter

Specifies that the rule is triggered by an EXEC statement. Used to validate the EXEC statement.

Optional. Type EXEC in the ON field and press Enter. The subparameters (selection Fields) shown in Table 31 are displayed.

Table 31  ON EXEC Subparameters

<table>
<thead>
<tr>
<th>Subparameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGM</td>
<td>The invoked program.</td>
</tr>
<tr>
<td>PROC</td>
<td>The invoked JCL procedure</td>
</tr>
<tr>
<td>O/N</td>
<td>Conjunctinal subparameter that permits linking of ON statements. Valid values are:</td>
</tr>
<tr>
<td>O/N</td>
<td>(Or) – Only one statement need be satisfied.</td>
</tr>
<tr>
<td>O/N</td>
<td>(Not) – The condition in the following statement need not be satisfied.</td>
</tr>
</tbody>
</table>

The JCL statement (EXEC card) variables that can be used with the ON EXEC statement are shown in Table 32.

Table 32  ON EXEC variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%%%$STEPNAME</td>
<td>Step name from the EXEC JCL statement</td>
</tr>
<tr>
<td>%%%$STEPNUM</td>
<td>Sequential number of the step in the JOB</td>
</tr>
<tr>
<td>%%%$EXECPROG</td>
<td>Program name</td>
</tr>
<tr>
<td>%%%$EXECPROC</td>
<td>Procedure name</td>
</tr>
<tr>
<td>%%%$EXECREGN</td>
<td>Step’s region size</td>
</tr>
<tr>
<td>%%%$EXECTIME</td>
<td>Step’s max CPU value</td>
</tr>
<tr>
<td>%%%$EXECParm</td>
<td>The PARAM value (up to 100 characters)</td>
</tr>
</tbody>
</table>

The user-defined variables that can be used with the ON EXEC statement are shown in Table 33.

Table 33  ON EXEC user-defined variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%%%EXMEMLMT</td>
<td>Step’s memory limit above the BAR</td>
</tr>
<tr>
<td>%%%EXPERFORM</td>
<td>Step’s WLM class</td>
</tr>
</tbody>
</table>
Example

In this example we check that no parameters are passed to the payment programs (note the use of O/N in the 2nd and 3rd sections).

Figure 50 ON EXEC Example

```
RULE DAILYPAY  ENVIRONMENT PROD  PRIORITY 03  CONTINUE SEARCH Y
DESC NO PARAMETERS ARE ALLOWED TO BE PASSED TO THE PAYMENT PROGRAMS

RULE YEARPAY  ENVIRONMENT PROD  PRIORITY 03  CONTINUE SEARCH Y
DESC NO PARAMETERS ARE ALLOWED TO BE PASSED TO THE PAYMENT PROGRAMS

JOBNAME PAY*
LIBRARY PRD.PAY.ACCOUNT.DAILY.JOBS  MEMBER DPay*
SCHLIB SCHDTAB  O/N O

JOBNAME PAY*
LIBRARY PRD.PAY.ACCOUNT.YEARLY.JOBS  MEMBER DPay*
SCHLIB SCHDTAB  O/N O

ON EXEC  PGM  PROC DAYPAY*  O/N O
ON EXEC  PGM  PROC YEARPAY*  O/N

IF  %E$EXCPARM NE ''
DO  MSG  TYPE E IT IS NOT PERMITTED TO PASS PARMS TO THESE PROGRAMS.
ENDIF
DO

============= >>>>>>>>>>>>>>> END OF JRLE DEFINITION PARAMETERS <<<<<<<<<<<<<<< =============
```
ON INCLUDE: Event Selection Parameter

Specifies that the rule is triggered by an INCLUDE statement. Used to validate the INCLUDE statements.

Optional. Type INCLUDE in the ON field and press Enter.

The following selection field (subparameter) is displayed:

MEMBER - The included member name

The following JCL statement (JOB card) variable can be used with the ON INCLUDE statement:

%%$INCLMEM - contains the name of the member to be included

Examples

Example 1

In this example, when a job contains an INCLUDE statement, the name of the member to be included is displayed in the output message.

Figure 51  INCLUDE example

<table>
<thead>
<tr>
<th>RULE TH015</th>
<th>ENVIRONMENT THMSD</th>
<th>PRIORITY 50</th>
<th>CONTINUE SEARCH Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESC</td>
<td>DESC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------</td>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>JOBNAME *</td>
<td>LIBRARY</td>
<td>MEMBER</td>
<td>SCHDTAB O/N</td>
</tr>
<tr>
<td>SCHDLIB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------</td>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>ON INCLUDE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------</td>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>DO MSG</td>
<td>TYPE I INCLMEM = %%$INCLMEM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DO</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ON JCLLIB: Event Selection Parameter

Specifies that the rule is triggered by a JCLLIB statement. Used to validate the JCLLIB statements.

Optional. Type JCLLIB in the ON field and press Enter.

The ON JCLLIB statement has the following subparameter:

- JCLLIB - The JCL library name.

The following JCL statement variable can be used with the ON JCLLIB statement:

- %%%$LIBDSNn - contains the dataset sequence number, according to the sequence that the datasets are defined in the JOB statements

- %%%$LIBDSN# - contains the total number of datasets defined in the JOB statements

Examples

Example 1

In this example, when a job contains a JCLLIB statement, the total number of datasets and the dataset sequence numbers are displayed in the output message.

Figure 52   JCLLIB example

<table>
<thead>
<tr>
<th>RULE TH012</th>
<th>ENVIRONMENT THMSB</th>
<th>PRIORITY 50</th>
<th>CONTINUE SEARCH Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESC</td>
<td>DESC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------</td>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>JOBNAME *</td>
<td>LIBRARY SCHDLIB</td>
<td>MEMBER SCHDTAB</td>
<td>O/N</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------</td>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>ON JCLLIB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------</td>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>DO MSG</td>
<td>TYPE I LIBDNS# = $$LIBDNS#</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DO MSG</td>
<td>TYPE I LIBDNS1 = $$LIBDNS1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DO MSG</td>
<td>TYPE I LIBDNS2 = $$LIBDNS2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DO</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ON JOB: Event Selection Parameter

Specifies that the rule is triggered by a JOB statement. Used to validate the JOB statement.

Optional. Type JOB in the ON field and press Enter. The subparameters (selection Fields) shown in Table 34 are displayed.

Table 34  ON JOB Subparameters

<table>
<thead>
<tr>
<th>Subparameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLASS</td>
<td>The job’s class.</td>
</tr>
<tr>
<td>PRTY</td>
<td>The job’s priority</td>
</tr>
<tr>
<td>USER</td>
<td>The job’s user ID</td>
</tr>
<tr>
<td>ACCOUNT</td>
<td>The job’s account</td>
</tr>
</tbody>
</table>
| O/N          | Conjunctonal subparameter that permits linking of ON statements. Valid values are:
|              | O (Or) – Only one statement need be satisfied. |
|              | N (Not) – The condition in the following statement need not be satisfied. |

The JCL statement (JOB card) variables that can be used with the ON JOB statement are shown in Table 35.

Table 35  ON JOB variables (part 1 of 2)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>% %$JOBNAME</td>
<td>The job name taken from the JOB card</td>
</tr>
<tr>
<td>% %$JOBACNT</td>
<td>The account information</td>
</tr>
<tr>
<td>% %$JOBPGMR</td>
<td>The programmer name</td>
</tr>
<tr>
<td>% %$JOBCLS</td>
<td>The JOB class value</td>
</tr>
<tr>
<td>% %$JOBPRTY</td>
<td>The JOB priority value</td>
</tr>
<tr>
<td>% %$JOBMSGC</td>
<td>The JOB msgclass value</td>
</tr>
<tr>
<td>% %$JOBMSGL</td>
<td>The JOB msglevel value</td>
</tr>
<tr>
<td>% %$JOBREGN</td>
<td>The JOB region size</td>
</tr>
<tr>
<td>% %$JOBTIME</td>
<td>The JOB max CPU value</td>
</tr>
<tr>
<td>% %$JOBUSER</td>
<td>The user name value</td>
</tr>
<tr>
<td>% %$JOBMEMLMT</td>
<td>The memory limit above the BAR</td>
</tr>
<tr>
<td>% %$JOBPERFORM</td>
<td>The WLM class for the job</td>
</tr>
<tr>
<td>% %$JOBRESTART</td>
<td>The JOB RESTART value</td>
</tr>
<tr>
<td>% %$JOBNOTIFY</td>
<td>The JOB NOTIFY value</td>
</tr>
</tbody>
</table>
Example

In this example, the job processing time for all jobs belonging to user EB38 are measured. If greater than 1000, the job processing time is displayed in the output message.

Table 35 ON JOB variables (part 2 of 2)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%%$JOBRD</td>
<td>The JOB RD value</td>
</tr>
<tr>
<td>%%$JOBADDRSPC</td>
<td>The JOB ADDRSPC value</td>
</tr>
</tbody>
</table>

Figure 53 ON JOB Example

In this example, the job processing time for all jobs belonging to user EB38 are measured. If greater than 1000, the job processing time is displayed in the output message.

```plaintext
+-----------------------------------------------------------------------------+
| RULE PRODXYZ     ENVIRONMENT GENERAL     PRIORITY 50    CONTINUE SEARCH Y   |
| DESC JOB PROCESSING TIME                                                    |
| DESC                                                                        |
|=============================================================================|
| JOBNAME *                                                                   |
| LIBRARY IOA.JCL                                      MEMBER  PRODXYZ        |
| SCHDLIB                                              SCHDTAB          O/N   |
|=============================================================================|
| ON JOB           CLASS         PRTY          USER EB38                      |
| ACCOUNT                                                               O/N   |
|=============================================================================|
| IF       %%$JOBTIME GT# 1000                                                |
| DO MSG      TYPE I THIS JOB TAKES %%$JOBTIME TO PROCESS.                    |
| DO                                                                          |
| ENDFI                                                                       |
| DO                                                                          |
+-----------------------------------------------------------------------------+  

<<<<<<<<< END OF JRLE DEFINITION PARAMETERS >>>>>>>>>>>>>>>>>>>>>>>>>>>
```
ON JOBEND: Event Selection Parameter

Specifies that the rule is triggered when a JCL job is ending. Used, for example, for cross JCL validations and counter checking.

Optional. Type JOBEND in the ON field and press Enter.

The ON JOBEND statement does not require any subparameters so there are no selection fields displayed.

All variables, of any type, with their latest values can be used with the ON JOBEND statement.

Examples

Example 1

In this example, when jobs are ending, their return and reason codes are displayed in the output message.

Figure 54   JOBINIT example

<table>
<thead>
<tr>
<th>RULE TH010</th>
<th>ENVIRONMENT THMSA</th>
<th>PRIORITY 50</th>
<th>CONTINUE SEARCH Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESC</td>
<td>DESC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOBNAME *</td>
<td>LIBRARY</td>
<td>MEMBER</td>
<td>SCHDTAB O/N</td>
</tr>
<tr>
<td>SCHDLIB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON JOBEND</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DO MSG</td>
<td>TYPE I RC = %$RC</td>
<td>RS = %$RS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ON JOBINIT: Event Selection Parameter

ON JOBINIT: Event Selection Parameter

Specifies that the rule is triggered when a JCL job is about to be processed. Used to initialize user variables (for example, counters).

Optional. Type JOBINIT in the ON field and press Enter.

The ON JOBINIT statement does not require any subparameters so there are no selection fields displayed.

The following JCL statement (JOB card) variable can be used with the ON JOBINIT statement:

%%$JOBNAME - The job name taken from the JOB card

Examples

Example 1

In this example, when a JCL job is about to be processed, information about the job is displayed in output messages.

Figure 55  JOBINIT example

<table>
<thead>
<tr>
<th>RULE STARTJOB1</th>
<th>ENVIRONMENT THM SA</th>
<th>PRIORITY 50</th>
<th>CONTINUE SEARCH Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESC</td>
<td>DESC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>JOBNAME M4918*</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIBRARY CTM.PROD.BKP(SPGBK)</td>
</tr>
<tr>
<td>SCHDLIB CTM.PROD.BKP(SPGBK)</td>
</tr>
<tr>
<td>SCHDTAB O/N</td>
</tr>
<tr>
<td>SCHDJOB</td>
</tr>
<tr>
<td>STATNUM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ON JOBINIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DO MSG TYPE I JCLTYPE = %%#JCLTYPE; MEMBER = %%#MEMBER</td>
</tr>
<tr>
<td>DO MSG TYPE I LIBRARY = %%#LIBRARY; ENVRN = %%#ENVRN</td>
</tr>
<tr>
<td>DO MSG TYPE I SCHDLIB = %%#SCHDLIB; SCHDTAB = %%#SCHDTAB</td>
</tr>
<tr>
<td>RETURN RC 0012 RS X' 0024 '</td>
</tr>
<tr>
<td>DO</td>
</tr>
</tbody>
</table>

124 BMC Control-M Workload Automation JCL Verify User Guide
ON PROC: Event Selection Parameter

Specifies that the rule is triggered by a procedure.

Optional. Type PROC in the ON field and press Enter.

The following selection field (subparameter) is displayed:

PROC – The procedure name

The following JCL statement (JOB card) variable can be used with the ON PROC statement:

% %$PROCNAME - contains the procedure name

Examples

Example 1

In this example, the name of the procedure is displayed in the output message.

Figure 56   JOBINIT example
RULE: General Parameter

Name of a rule as specified in the entry panel. The name must be unique in the rule table, and therefore you cannot define two rules in the same table with the same rule name.

General Information

The RULE parameter does not operate as a selection parameter. It serves to identify the rule.

Example

In this example, the rule named DAILYPAY checks that no parameters are passed to the payment programs.

Figure 57 RULE Example

```
+-----------------------------------------------------------------------------+
| RULE DAILYPAY   ENVIRONMENT PROD       PRIORITY 3 CONTINUE SEARCH Y    |
| DESC        No parameters are allowed to be passed to the payment programs  |
|=============================================================================|
| JOBNAME PAY*____            _                                               |
| LIBRARY  PRD.PAY.ACCOUNT.DAILY.JOBS__________________  MEMBER  DPAY*___     |
| SCHDLIB                                              SCHDTAB          O/N   |
| JOBNAME PAY*____            _                                               |
| LIBRARY  PRD.PAY.ACCOUNT.YEARLY.JOBS__________________ MEMBER  DPAY*___     |
| SCHDLIB                                              SCHDTAB          O/N   |
|=============================================================================|
| ON EXEC     PGM          PROC DAYP*                                   O/N O |
| ON EXEC     PGM          PROC YEARP*                                  O/N   |
|=============================================================================|
| IF %%$EXECPARM NE ''                                                        |
| DO MSG  TYPE E It is not permitted to pass parms to these programs.         |
| ENDIF                                                                       |
|=============================================================================|
+====== >>>>>>>>>>>>>>> END OF RULE DEFINITION PARAMETERS <<<<<<<<<<<<<<<< ====
```
SCHDLIB: Common Selection Parameter

Name of the scheduling library when invoked from a Control-M/EM interface.

General Information

This optional selection parameter is common to all the various ON parameter types.

Examples

Example 1

In this example, information about the jobs in the Control-M scheduling library, CTM.PROD.BKP(SPBKP), that are about to be processed, are displayed in output messages.

Figure 58  SCHDLIB example

```
+-----------------------------------------------------------------------------+
| RULE STARTJOB1   ENVIRONMENT THMSA       PRIORITY 50    CONTINUE SEARCH Y   |
| DESC                                                                        |
| DESC                                                                        |
|=============================================================================|
| JOBNAME M4918*                                                              |
| LIBRARY                  MEMBER                     |
| SCHDLIB CTM.PROD.BKP(SPBKP)                          SCHDTAB          O/N   |
|=============================================================================|
| ON JOBINIT                                                                  |
|=============================================================================|
| DO MSG      TYPE I JCLTYPE = %%#JCLTYPE; MEMBER  = %%#MEMBER                |
| DO MSG      TYPE I LIBRARY = %%#LIBRARY; ENVRN   = %%#ENVRN                 |
| DO MSG      TYPE I SCHDLIB = %%#SCHDLIB; SCHDTAB = %%#SCHDTAB               |
| DO MSG      TYPE I SCHDJOB = %%#SCHDJOB; STATNUM = %%#STATNUM               |
| RETURN   RC 0012 RS X' 0024 '                                               |
| DO                                                                          |
|=============================================================================|
```
SCHTAB: Common Selection Parameter

Name of the scheduling table when invoked from a Control-M/EM interface. (Note that the name does not refer to “folder” since “table” is still the term used with Control-M for z/OS.)

General Information

This optional selection parameter is common to all the various ON parameter types.

Examples

Example 1

In this example, information about the jobs in the Control-M scheduling table, PAYROLL, that are about to be processed, are displayed in output messages.

Figure 59  JOBINIT example

```
+-----------------------------------------------------------------------------+
| RULE STARTJOB1   ENVIRONMENT THMSA       PRIORITY 50    CONTINUE SEARCH Y   |
| DESC                                                                        |
| DESC                                                                        |
|=============================================================================|
| JOBNAME M4918*                                                              |
| LIBRARY                                              MEMBER                 |
| SCHDLIB                                              SCHDTAB PAYROLL  O/N   |
|=============================================================================|
| ON JOBINIT                                                                  |
|=============================================================================|
| DO MSG      TYPE I JCLTYPE = %%#JCLTYPE; MEMBER  = %%#MEMBER                |
| DO MSG      TYPE I LIBRARY = %%#LIBRARY; ENVRN   = %%#ENVRN                 |
| DO MSG      TYPE I SCHDLIB = %%#SCHDLIB; SCHDTAB = %%#SCHDTAB               |
| DO MSG      TYPE I SCHDJOB = %%#SCHDJOB; STATNUM = %%#STATNUM               |
| RETURN   RC 0012 RS X' 0024 '                                               |
| DO                                                                          |
|=============================================================================|
```
Job flow verification (CTJMRFLW utility)

In addition to single job verification, Control-M/JCL Verify can validate work flows through the verification of the job order and dependencies. Control-M/JCL Verify can verify a mixture of Control-M schedule definition and regular JCL jobs.

A procedure, named CTJMRFLW, invokes Control-M to determine the job order and job dependencies. Then, after arranging the jobs and schedule definitions in the order as they are intended to run, CTJMRFLW invokes Control-M/JCL Verify for the job flow verification. At each stage of the job flow, Control-M/JCL verify examines the datasets of the previously verified jobs to determine if those datasets still exist or were deleted, and, based on the dependencies, verifies the job flow.

The CTJMRFLW can manage up to 10,000 accumulated datasets that are used by all the jobs that are verified in the execution. The limitation in this utility is not based on the maximum jobs number, but from the number of all the datasets that are processed by all the jobs in the execution.

To verify the current job’s datasets, the status of the datasets are determined by the following factors:

- the entire job flow hierarchy to which the job belongs
- the previous instances of the datasets
- all the verified jobs at the parallel hierarchic level, whether they are directly or indirectly related to the job

In some situations, where the results of the dependence analysis might be ambiguous, warning messages are displayed and the user must investigate the situation.
For example, if two jobs are both directly dependent on a previous job, by definition these two jobs will run on the same level of hierarchy, which is lower than the previous job. Even though it is certain that these two jobs will run after the higher-level job, it is not certain in what order the two equal-level jobs will run. The warning messages alert the user to investigate whether the actual order of the jobs will be problematic or not.

--- NOTE ---

Even though that Control-M arranges the JCLs for Control-M/JCL Verify, there is a chance that datasets will be allocated or handled besides these JCLs.

This chapter includes the following topics:

- Parameters ................................................................. 131
- Activating the Utility ................................................... 133
- Batch JCL .................................................................. 133
- Return Codes .............................................................. 133
- Example .................................................................... 134
Table 36 describes the parameters that are used for the CTJMRFLW procedure:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| ML        | The minimum level of message severity to be issued. Valid values are:  
|           | ■ I – Information – all messages are issued.  
|           | ■ W – warning messages and errors are issued.  
|           | ■ E – Only error messages are issued  
|           | ■ D - as defined by the MSGLEVEL parameter in the CTJPARM member. Default |
| FA        | Whether to verify file access privileges. Valid values are:  
|           | ■ Y - Yes  
|           | ■ N - No  
|           | ■ D - as defined by the DSNACCSS parameter in the CTJPARM member. Default |
| FE        | Whether to verify file existence. Valid values are:  
|           | ■ Y - Yes  
|           | ■ N - No  
|           | ■ D - as defined by the DSNEXIST parameter in the CTJPARM member. Default |
| ME        | Whether to verify that the load modules exist. Valid values are:  
|           | ■ Y - Yes  
|           | ■ N - No  
|           | ■ D - as defined by the PGMCHECK parameter in the CTJPARM member. Default |
| STDR      | Whether to use the Control-M/JCL Verify rules to verify site standards. Valid values:  
|           | ■ Y - Yes  
|           | ■ N - No  
|           | ■ D - as defined by the STDR parameter in the CTJPARM member. Default |
| ENV       | Whether to use the rules that are defined for the specified environment, in addition to the GENERAL rules, for the site standard verifications. If the keyword GENERAL is specified, only the GENERAL rules are used for the site standard verifications. |
Table 37 describes the keywords used in the SYSIN (or JOBLIST.DAJOB) statements to perform various verification tasks. The table indicates which keywords are used for each task.

Table 37 CTJMRFLW JOBLIST.DAJOB statement keywords

<table>
<thead>
<tr>
<th>Use these keywords:</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORDER</td>
<td>Defines a new statement. It is used only for simulation.</td>
</tr>
<tr>
<td>DSN</td>
<td>Schedule library name where the JCL jobs to be verified are located. Mandatory.</td>
</tr>
<tr>
<td>MEMBER</td>
<td>Member name where the Control-M Schedule table containing the JCL jobs to be verified is located. Character masking (* &amp; %) is supported so that “MEMBER=*” indicates that all the jobs in the library are to be verified.</td>
</tr>
<tr>
<td>JOB</td>
<td>Name of the job scheduling definition to be verified. Character masking (* &amp; %) is supported so that “JOB=*” indicates that all the jobs in the table are to be verified. Optional.</td>
</tr>
</tbody>
</table>
Activating the Utility

You can only activate the utility through a batch JCL.

Batch JCL

The following is a sample batch JCL used to invoke CTJMRFLW:

```
//VVERFLW EXEC CTJMRFLW,FA=D,ENV=GENERAL
//JOBLIST,DAJOB DD *
  ORDER DSN=CTMP.V800.OPR.SCHEDULE, MEMBER=TEST, JOB=TESTVAR2
//
```

When FA, FE and other parameters are not specified in the EXEC statement, CTJMRFLW uses the default values according to the batch rules.

Return Codes

The return codes for the CTJMRFLW are explained in Table 3 in “Return Codes” on page 34.
In the following sample job, located at `ilprefa.CTJ.JCL(CTJJRFLW)`, `CTJMRF LW` verifies the job dependencies of the series of JCL jobs according to the order that they are intended to run.

**Figure 60  Dependency awareness– Example 1**

```plaintext
//I800INVD JOB ,IOA800,MSGCLASS=X,CLASS=A
//*
//*
//  JCLLIB ORDER=IOAE.R800TEST.PROCLIB
//  INCLUDE MEMBER=IOASET
//******************************************************************************
//**
//**   PARAMETERS TO CTJMRF LW
//**      ML=D/I/W/E  I=INFO W=WARNING E=ERROR
//**      FA=D/Y/N
//**      FE=D/Y/N
//**      ME=D/Y/N
//**      UT=D/Y/N
//**      STOR=D/Y/N
//**      D=DEFAULT (SET IN CTJPARM) Y=YES N=NO
//**
//**      ENV=GENERAL
//**
//**      ODATE=CONTROL-M WORKING DATE
//**
//**      EMUSRDLY=EMUSRDLY, EM SYSTEM NAME LIST MEMBER
//**
//**      AJFSZ=1000, EQUAL TO AJF SIZE
//**
//**      CHARSET=GS10, STANDARD ENGLISH CHARACTER SET
//**
//**      CHARST2=FM10, CHAR-SET CONTAINS BOX ELEMENTS
//**
//**      OUT='*' GENERAL SYSOUT CLASS
//**
//**  STATEMENTS
//**
//**      ORDER DEFINE NEW STATEMENT,IT IS ONLY FOR SIMULATION
//**
//**      DSN= SCHEDULE LIBRARY (MANDATORY)
//**
//**      MEMBER= NAME OF CTM SCHED TABLE SUPPORT MASKING (* & %)
//**
//**      JOB= JOB IS OPTIONAL AND SUPPORT MASKING (* & %)
//**
//**
//**  REFER TO CTJPLAN UTILITY CHAPTER IN CONTROL-M/JCL VERIFY
//**
//**  USER GUIDE
//**
//******************************************************************************
/JVERFLW EXEC CTJMRF LW
//JOBLIST.DAJOB DD *
  ORDER DSN=CTM.V000.OPR.SCHEDULE MEMBER=MAINDAY <- CHANGE THIS
```
The CTJPLAN utility is designed as part of a procedure that is capable of verifying JCL jobs that are scheduled for submission. The procedure requires that Control-M is installed. The procedure involves using the CTMRPLN and CTJVER utilities.

The procedure can verify jobs referred to from Control-M job definitions that will be ordered on the specified date. The procedure can either verify a single job or work in mass mode, verifying many jobs together (for example, verifying all the jobs in a library).

This chapter includes the following topics:

- Procedure for verifying scheduled JCL jobs .......................................................... 136
- Parameters .............................................................................................................. 136
- Activating the Utility ............................................................................................ 138
  - Batch JCL ........................................................................................................... 138
- Return Codes ........................................................................................................ 139
- CTJPLAN output report ....................................................................................... 139
- Examples ............................................................................................................... 139
  - Example 1 .......................................................................................................... 139
Procedure for verifying scheduled JCL jobs

To verify JCL jobs that are scheduled for submission

Run the CTJPLAN job, which consists of the following two steps:

- CTMRPLN gets the list of all the jobs that are scheduled for a specified ODATE. CTMRPLN creates the list of jobs as valid statements for CTJVER.

- CTJVER processes the scheduled jobs using the SYSIN statements generated by CTMRPLN in the preceding step.

**NOTE**
If MULTJOBS=Y is specified in the CTMPARM member, only the first job will be submitted from the JCL member by Control-M. Therefore, if the JCL member to be verified contains more than one job, only the first job will be verified.

Parameters

Table 38 describes the parameters that the utility receives from the JCL procedure:

**Table 38**  
CTJPLAN PROCEDURE parameters (part 1 of 3)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODATE</td>
<td>Original scheduling date of the job, in <em>ymmd</em> format.</td>
</tr>
<tr>
<td>JES</td>
<td>Whether to verify that the JES2 or JES3 statements are correct. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>- Y - Yes</td>
</tr>
<tr>
<td></td>
<td>- N - No</td>
</tr>
<tr>
<td></td>
<td>- D - as defined by the JESTTMNT parameter in the CTJPARM member. Default</td>
</tr>
<tr>
<td>FE</td>
<td>Whether to verify file existence. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>- Y - Yes</td>
</tr>
<tr>
<td></td>
<td>- N - No</td>
</tr>
<tr>
<td></td>
<td>- D - as defined by the DSNEXIST parameter in the CTJPARM member. Default</td>
</tr>
</tbody>
</table>
### Table 38  CTJPLAN PROCEDURE parameters (part 2 of 3)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FA</strong></td>
<td>Whether to verify file access privileges. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>■ Y - Yes</td>
</tr>
<tr>
<td></td>
<td>■ N - No</td>
</tr>
<tr>
<td></td>
<td>■ D - as defined by the DSNACCSS parameter in the CTJPARM member. Default</td>
</tr>
<tr>
<td><strong>STDR</strong></td>
<td>Whether to use the Control-M/JCL Verify rules to verify site standards. Valid values:</td>
</tr>
<tr>
<td></td>
<td>■ Y - Yes</td>
</tr>
<tr>
<td></td>
<td>■ N - No</td>
</tr>
<tr>
<td></td>
<td>■ D - as defined by the STDR parameter in the CTJPARM member. Default</td>
</tr>
<tr>
<td><strong>ENV</strong></td>
<td>Whether to use the rules that are defined for the specified environment, in addition to the GENERAL rules, for the site standard verifications. If the keyword GENERAL is specified, only the GENERAL rules are used for the site standard verifications.</td>
</tr>
<tr>
<td><strong>ME</strong></td>
<td>Whether to verify that the load modules exist. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>■ Y - Yes</td>
</tr>
<tr>
<td></td>
<td>■ N - No</td>
</tr>
<tr>
<td></td>
<td>■ D - as defined by the PGMCHECK parameter in the CTJPARM member. Default</td>
</tr>
<tr>
<td><strong>ML</strong></td>
<td>The minimum level of message severity to be issued. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>■ I – Information – all messages are issued.</td>
</tr>
<tr>
<td></td>
<td>■ W – warning messages and errors are issued.</td>
</tr>
<tr>
<td></td>
<td>■ E – Only error messages are issued</td>
</tr>
<tr>
<td></td>
<td>■ D - as defined by the MSGLEVEL parameter in the CTJPARM member. Default</td>
</tr>
<tr>
<td><strong>CTM</strong></td>
<td>Whether to resolve Control-M AutoEdit variables in the job. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>■ Y - Yes</td>
</tr>
<tr>
<td></td>
<td>■ N - No</td>
</tr>
<tr>
<td></td>
<td>■ D - as defined by the CTMVAR parameter in the CTJPARM member. Default</td>
</tr>
</tbody>
</table>
Activating the Utility

You can activate the utility through a batch JCL.

Batch JCL

The following is a sample batch JCL used to invoke CTJPLAN:

```apl
// .... JOB ....
//      EXEC CTJPLAN, ODATE=odate
/*

In the sample, the scheduling library is the Control-M installation scheduling library.

To specify another library, override the SCHEDLIB DD statement. For example, add
the following statement:

```apl
//SCHEDLIB DD DISP=SHR, DSN=MY.CTM.SCHEDLIB
```

When FA, FE and other parameters are not specified in the EXEC statement, CTJPLAN uses the default values according to the batch rules.

To specify an individual or multiple tables, add a SYSIN statement with the TABLES parameter. For example, adding the following statement will include all tables with either a abc or xyz prefix:

```apl
// .... SUB ....
/*
// TABLES abc*, xyz*

Note: UT=Y must be used with ME=Y, FE=Y, and FA=Y.
```

UT Whether to verify the syntax of the IEBGENER, IEBCOPY, and SORT IBM utilities. Valid values are:

- Y - Yes
- N - No
- D - as defined by the SUPUTIL parameter in the CTJPARM member. Default

Table 38 CTJPLAN PROCEDURE parameters (part 3 of 3)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UT</td>
<td>Whether to verify the syntax of the IEBGENER, IEBCOPY, and SORT IBM utilities. Valid values are:</td>
</tr>
<tr>
<td>TABLES abc*, xyz*</td>
<td>Enables use of generic table names to process individual or multiple tables. Use with the SYSIN statement, as shown in “Batch JCL” on page 138. Optional.</td>
</tr>
</tbody>
</table>
Return Codes

The return codes for the RPLAN step are explained in the CTMRPLN utility in the INCONTROL for z/OS Utilities Guide.

The return codes for the CTJVER JVER step are explained in Table 3 in “Return Codes” on page 34.

CTJPLAN output report

The report includes

- the JOB definitions listing as selected by the CTMRPLN utility
- the JCL listing as received from z/OS
- the messages issued by z/OS messages issued by JVER which are separated into 3 severity types: I-Information, W-Warning, and E-Error.

The user can choose the minimum level of messages to be issued using the CTJPARM parameter MSGLEVEL, which can be overridden from the interfaces. These messages are issued only after the job has passed the valid syntax check.

Examples

Example 1

In the following example, CTJPLAN verifies all jobs that are planned to be executed on June 6, 2013.

The selection is performed on the first step of the CTJPLAN procedure, which creates a list of scheduling libraries, members, and jobs for verification.

The verification is performed in the second step.
The verification parameters are the same as in CTJVER utility and in the above sample use the defaults.

In the first step, the CTMRPLN utility creates DAREPORT, which contains the list of the jobs that are planned to be executed on June 6, 2013.

Figure 62  CTJPLAN - CTMRPLN planned JOBS report

Figure 63 shows the output report job (in slightly abbreviated form) for the jobs in the list shown in Figure 62 above.

Please note the following about the verification report shown in Figure 63:

- The RC=08 resulted because not all JCL members contained valid JCL JOB statements.
- Each job is called separately from CTJVER utility using the following code:

```
SCHEDLIB=ControlM.SCHEDULE.Library  TABLE=table  +
JOB=jobname
```

The “+” (appearing in column 72) allows the input to continue with JOB=jobname on the next CTJVER statement.
When the CTJPLAN was executed, the shared table was not initialized and therefore the CTJ202I message was issued.

**Figure 63  CTJPLAN - CTJVER verification report (part 1 of 3)**

<table>
<thead>
<tr>
<th>CTJU01I</th>
<th>FA=D FE=D JES=D ML=D ML=D</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTJU01I</td>
<td>ODATE=120606</td>
</tr>
<tr>
<td>CTJU01I</td>
<td>SCHEDLIB=IOAZ.CTJDR3.CTM.OPR.SCHEDULE TABLE=MAINDAY +</td>
</tr>
<tr>
<td>CTJU01I</td>
<td>JOB=DAILYPRD</td>
</tr>
</tbody>
</table>

CTJ202I Shared INFO not found. Run CTJINIT to improve verification response time

CTJ000I Control-M/JCL Verify is starting. Level IJ100B6

CTJ00CI Runtime parameters: FA=Y FE=Y STOR=Y ENV=GENERAL JES=Y ME=Y ML=I USER=N18

CTJR08I Verify JCL in JOB DAILYPRD from TABLE MAINDAY in LIBRARY IOAZ.CTJDR3.CTM.OPR.SCHEDULE

CTJ002I Start verifying MEMBER MAINDAY DSNAME=CTMP.V700.JCL

CTJR05I //I7001NDP JOB .IOA700,MSGCLASS=X,CLASS=A

CTJ008I Start verifying JOB I700INDP MEMBER MAINDAY

-\[\text{0010000}^{00010000}^{00020000}^{00030000}^{00040000}^{00050000}^{00060000}^{00070000}^{00080000}^{00090000}^{00100000}\]

CTJE002I INCLUDE GROUP IOASET WAS EXPANDED USING PRIVATE LIBRARY IOAP.V700.PROCLIB

CTJE001I PROCEDURE CTMDAILY WAS EXPANDED USING PRIVATE LIBRARY IOAP.V700.PROCLIB

XXCTMDAILY PROC REG='0M', DEFAULT REGION

XX PROGLST=PROGUSR, PROGRAMS ACTIVATED ON DAILY

XX DATEREC=, USER DATEREC

XX* EMUSRDLY=EMUSRDLY, EM SYSTEM NAME LIST MEMBER

XX OUT='*' GENERAL SYSOUT CLASS

XX*

75 XXUSRDAILY EXEC PGM=CTMLU,REGION=&REG

IEFC653I SUBSTITUTION JCL - PGM=CTMLU,REGION=OM

76 XX INCLUDE MEMBER=IOAENV

IEFC653I SUBSTITUTION JCL - MEMBER=IOAENV

77 XXSTEPPLIB DD DISP=SHR,DSN=IOAP.V700.TLOAD

78 XX DD DISP=SHR,DSN=CTMLU

IEFC653I SUBSTITUTION JCL - DISP=SHR,DSN=IOAP.V700.TLOAD

79 XX DD DISP=SHR,DSN=IOAP.V700.TLOADE

80 XX DD DISP=SHR,DSN=CTMLU

IEFC653I SUBSTITUTION JCL - DISP=SHR,DSN=IOAP.V700.TLOAD

81 XX DD DISP=SHR,DSN=&CTRANS

IEFC653I SUBSTITUTION JCL - DISP=SHR,DSN=IOAP.V700.CTRANS

82 XXDAPARM DD DISP=SHR,DSN=ILPREFA..PARM

IEFC653I SUBSTITUTION JCL - DISP=SHR,DSN=CTMLU

83 XX DD DISP=SHR,DSN=ILPREFA..IOAENV

IEFC653I SUBSTITUTION JCL - DISP=SHR,DSN=IOAP.V700.IOAENV

XX*
Figure 63  CTJPLAN - CTJVER verification report  (part 2 of 3)

84  XXDAALOCIN DD DISP=SHR,DSN=IOAALOCIN(ALCMUADAY)
    IEFCS65I SUBSTITUTION JCL - DISP=SHR,DSN=IOAP.V700.IOAENV(ALCMUADAY)
85  XXDAPROG DD DISP=SHR,DSN=OLPREFM..PARM(ProgrlSt)
    IEFCS65I SUBSTITUTION JCL - DISP=SHR,DSN=CTMP.V700.0PR.PARM(PROGUSR)
86  XXDACCHK DD DISP=SHR,DSN=OLPREFM..PARM(DATEREC)
    IEFCS65I SUBSTITUTION JCL - DISP=SHR,DSN=CTMP.V700.0PR.PARM(DATERECU)
87  XXDAOUT DD SYSOUT=*
    IEFCS65I SUBSTITUTION JCL - SYSOUT=*  88  XXSYSPRINT DD SYSOUT=*
    IEFCS65I SUBSTITUTION JCL - SYSOUT=*
89  XXSYSABEND DD SYSOUT=OUTDUMP
    IEFCS65I SUBSTITUTION JCL - SYSOUT=X
90  //DAJOB DD DISP=SHR,DSN=OLPREFM..SCHEDULE(TABLE1)  (CHANGE 00110001
    IEFCS65I SUBSTITUTION JCL - DISP=SHR,DSN=CTMP.V700.SCHEDULE(TABLE1)
    CTJDO0W 90 WARNING: MEMBER=TABLE1 IS NOT FOUND AT DSN=CTMP.V700.SCHEDULE
91  // DD DISP=SHR,DSN=OLPREFM..SCHEDULE(TABLE2)  (CHANGE 00120001
    IEFCS65I SUBSTITUTION JCL - DISP=SHR,DSN=CTMP.V700.SCHEDULE(TABLE2)
    CTJDO0W 91 WARNING: MEMBER=TABLE2 IS NOT FOUND AT DSN=CTMP.V700.SCHEDULE
92  // 00130000
    CTJ009I Processing ended RC=0004 for JOB I700INDP MEMBER MAINDAY
    00130000
    CTJ003I Processing ended RC=0004 REASON 00000000 MEMBER MAINDAY DSNAME=CTMP.V700.JCL
    CTJU01I SCHEDLIB=IOAZ.CTJDR3.CTM.OPR.SCHEDULE TABLE=MAINDAY +
    CTJU01I JOB=IOACLCND
    CTJ00CI Runtime parameters: FA=Y FE=Y STD=Y ENV=GENERAL JES=Y ME=Y ML=I USER=N18
    CTJR08I Start verifying JCL in JOB DAILYSYS from TABLE MAINDAY in LIBRARY IOAZ.CTJDR3.CTM.OPR.SCHEDULE
    CTJ002I Start verifying MEMBER MAINDAY DSNAME=CTMP.V700.JCL
    //:666666666666666666
    CTJS03E ERROR: Invalid or missing JOB statement
    /*NET ID=AESUSER
    /*----- SUBMITTED BY CONTROL-M (FROM MEMLIB) ODATE=120702
    //
    CTJ003I Processing ended RC=0008 REASON 00000000 MEMBER MAINDAY DSNAME=CTMP.V700.JCL
    CTJU01I SCHEDLIB=IOAZ.CTJDR3.CTM.OPR.SCHEDULE TABLE=MAINDAY +
    CTJU01I JOB=IOACLDND
    CTJ00CI Runtime parameters: FA=Y FE=Y STD=Y ENV=GENERAL JES=Y ME=Y ML=I USER=N18
    CTJ002I Start verifying MEMBER MAINDAY DSNAME=IOAP.V700.JC
    SUB136E NO JOB CARD
    /*xdjfbglzdkgdkgkdjng
    CTJS03E ERROR: Invalid or missing JOB statement
    /*NET ID=AESUSER
    /*----- SUBMITTED BY CONTROL-M (FROM MEMLIB) ODATE=120702
    //
    CTJ003I Processing ended RC=0008 REASON 00000000 MEMBER MAINDAY DSNAME=CTMP.V700.JCL
    CTJU01I SCHEDLIB=IOAZ.CTJDR3.CTM.OPR.SCHEDULE TABLE=MAINDAY +
    CTJU01I JOB=IOALDNRS
    CTJ00CI Runtime parameters: FA=Y FE=Y STD=Y ENV=GENERAL JES=Y ME=Y ML=I USER=N18
    CTJR08I Verify JCL in JOB IOALDNRS from TABLE MAINDAY in LIBRARY IOAZ.CTJDR3.CTM.OPR.SCHEDULE
    CTJ002I Start verifying MEMBER MAINDAY DSNAME=IOAP.V700.JC
    SUB136E NO JOB CARD
    CTJR05I  //I700INLD JOB .IOA700,MSGCLASS=X,CLASS=A,NOTIFY=K60,
    CTJ008I Start verifying JOB I700INLD MEMBER MAINDAY
    1 //I700INLD JOB .IOA700,MSGCLASS=X,CLASS=A,NOTIFY=K60,
    // MSGLEVEL=(1,1),
    // USER=PRODMMGR
    /*NET ID=AESUSER
    /*----- SUBMITTED BY CONTROL-M (FROM MEMLIB) ODATE=120702
    /*----- SCHEDULE (UNKNOWN)
    /*----- SCHEDULE DUE TO RBC:
Figure 63  CTJPLAN - CTJVER verification report  (part 3 of 3)

```c
/*---- JCL  IOAP.V700.JCL(IOALDNRS)
/*---- CONTROL-M JOB IDENTIFICATION:  ORDER ID=00012  RUN NO.=00001
/*
/*
2 //   JCLLIB ORDER=IOAP.V700.PROCLIB  00003000
3 //   INCLUDE MEMBER=IOASET
3 IEFC002I INCLUDE GROUP IOASET WAS EXPANDED USING PRIVATE LIBRARY IOAP.V700.PROCLIB
73 //LOADMAN EXEC IOALDNRS  00070001
73 IEFC001I PROCEDURE IOALDNRS WAS EXPANDED USING PRIVATE LIBRARY IOAP.V700.PROCLIB
75 XXIOALDNRS EXEC PGM=CTMLNR,REGION=&REG,PARM='ALL'  00120000
75 IEFC653I SUBSTITUTION JCL - PGM=CTMLNR,REGION=OM,PARM="ALL"
76 CTJ009I INFORMATION: PGM CTMLNR FOUND IN STEPLIB LIBRARY IOAP.V700.LOAD
76 XX INCLUDE MEMBER=&IOAENV  00130000
76 IEFC653I SUBSTITUTION JCL - MEMBER=IOAENV
77 XXSTEPLIB DD DISP=SHR,DSN=IOAP.V700.TLOAD
78 XX DD DISP=SHR,DSN=&STEPLIB
78 IEFC653I SUBSTITUTION JCL - DISP=SHR,DSN=IOAP.V700.TLOAD,DSNAME=IOAENV
79 XX DD DISP=SHR,DSN=IOAP.V700.TLOADE
80 XX DD DISP=SHR,DSN=&STEPLIBE
80 IEFC653I SUBSTITUTION JCL - DISP=SHR,DSN=IOAP.V700.TLOADE
77 IEFC653I SUBSTITUTION JCL - DISP=SHR,DSN=IOAP.V700.TLOAD,DSNAME=IOAENV
81 XX DD DISP=SHR,DSN=&CTRANS
81 IEFC653I SUBSTITUTION JCL - DISP=SHR,DSN=IOAP.V700.CTRANS
82 XXDAPARM DD DISP=SHR,DSN=&ILPREFA..PARAM
82 IEFC653I SUBSTITUTION JCL - DISP=SHR,DSN=IOAP.V700.PARM
83 XX DD DISP=SHR,DSN=&ILPREFA..IOAENV
83 IEFC653I SUBSTITUTION JCL - DISP=SHR,DSN=IOAP.V700.IOAENV
84 XXSORTIN DD DSN=&&SORTIN,UNIT=&WORKUNIT,SPACE=(CYL,(1,1))  00140000
84 IEFC653I SUBSTITUTION JCL - DSN=&&SORTIN,UNIT=SYSALLDA,SPACE=(CYL,(1,1))
85 XXSORTOUT DD DSN=&&SORTOUT,UNIT=&WORKUNIT,SPACE=(CYL,(1,1)),DCB=(LRECL=49)  00150000
85 IEFC653I SUBSTITUTION JCL - DSN=&&SORTOUT,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),DCB=(LRECL=49)
86 XXSORTWK01 DD UNIT=&WORKUNIT,SPACE=(CYL,(1,1))  00170000
86 IEFC653I SUBSTITUTION JCL - UNIT=SYSALLDA,SPACE=(CYL,(1,1))
87 XXSORTWK02 DD UNIT=&WORKUNIT,SPACE=(CYL,(1,1))  00180000
87 IEFC653I SUBSTITUTION JCL - UNIT=SYSALLDA,SPACE=(CYL,(1,1))
88 XXSORTWK03 DD UNIT=&WORKUNIT,SPACE=(CYL,(1,1))  00190000
88 IEFC653I SUBSTITUTION JCL - UNIT=SYSALLDA,SPACE=(CYL,(1,1))
89 XXSYSPRINT DD SYSPRTOUT=OUTDUMP
89 IEFC653I SUBSTITUTION JCL - SYSOUT=OUTDUMP
90 XXSYSOUT DD SYSOUT=OUTDUMP
90 IEFC653I SUBSTITUTION JCL - SYSOUT=OUTDUMP
91 XXSYSABEND DD SYSOUT=OUTDUMP
91 IEFC653I SUBSTITUTION JCL - SYSOUT=OUTDUMP
92 XXPRTOB6 DD SYSOUT=OUTDUMP
92 IEFC653I SUBSTITUTION JCL - SYSOUT=OUTDUMP
93 XXDAALOCIN DD DISP=SHR,DSN=IAALOCIN(ALCLDNRS)
93 IEFC653I SUBSTITUTION JCL - DISP=SHR,DSN=IOAP.V700.IOAENV(ALCLDNRS)
94 XXDALNRIN DD DONAME=SYSIN
94 IEFC653I SUBSTITUTION JCL - DONAME=SYSIN
95 WARNING 00020000
95 IEF668I DDNAME REFERRED TO ON DDNAME KEYWORD IN PRIOR STEP WAS NOT RESOLVED
96 CTJ009I Processing ended RC=0004 for JOB I700INLD MEMBER MAINDAY
96 CTJ003I Processing ended RC=0004 REASON 00000000 MEMBER IOALDNRS DSNAME GENERAL
CTJU02I Control-M/JCL Verify Utility ended. RC=0008
```
CTJXVER edit macro

The CTJXVER edit macro, which is capable of verifying a JCL job before it is submitted, can be invoked from the ISPF editor.

This chapter includes the following topics:

Overview ................................................................. 145
Syntax ................................................................. 146
   Verification mode .................................................. 146
Return Codes ......................................................... 147
Examples .............................................................. 148
   Example 1 ......................................................... 148

Overview

The CTJXVER edit macro can be used to verify JCL jobs before they are submitted. The CTJXVER edit macro can be invoked from the ISPF editor or from any other compatible editor. In an Edit or View session of the job member, enter the CTJXVER macro on the command line or use it as a line command.

NOTE

BMC Software ships the edit macro as CTJXVER, but the macro can be copied to the SYSPROC library as JVER, making it easier to run. For details, refer to the INCONTROL for z/OS Installation Guide: Installing, Step 23.5 - Edit Macro Customization.

The CTJXVER edit macro works with the member in memory, so any change while editing the member will be verified even if the member is not saved.
After processing, the CTJXVER edit macro displays the results in the job member. The displayed results is a simulation of the error messages that would be displayed if the job actually ran. The display is divided into the following two sections:

- header
- main body

The header section includes a summary of the number and types of messages.

The main body section contains the job member with the messages appearing immediately after the lines in the job where they are relevant.

### Syntax

The CTJXVER edit macro has the following syntax:

```plaintext
CTJXVER [?] [P]
```

The CTJXVER edit macro can be invoked in one of the following modes:

- verification mode
- message filter mode

### Verification mode

In verification mode, the CTJXVER edit macro uses the following syntax:

```plaintext
CTJXVER [P]
```

When CTJXVER is invoked without the P option, the verification tests are performed using the latest default parameters.

--- **NOTE**

CTJXVER takes the value of VERIFICATION CRITERIA from the User IOA Profile or the site default profile member which is in the IOA Profile library (not the ISPF Profile library).

The user must enter the JD screen in the IOA online environment to set the defaults (if needed). The user profile is saved only when the user exits the IOA online environment.
To set new default parameters, invoke CTJXVER with the P option. A window opens (see Figure 64) allowing you to set the new parameters, before the verification.

Figure 64  Control-M/JCL Verify Definition Facility

![Control-M/JCL Verify Definition Facility](image-url)

Table 39  CTJXVER Edit Macro Return Codes (part 1 of 2)

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Operation performed successfully.</td>
</tr>
<tr>
<td>4</td>
<td>Control-M/JCL Verify discovered that the verified resource is not in the required status at the time of the verification. For example, an input dataset does not exist. Review the status and decide if there is a need to make a change in the job.</td>
</tr>
<tr>
<td>8</td>
<td>Control-M/JCL Verify discovered a problem in the job that will cause it to fail with a JCL error or to be rejected by JES2 or JES3, either during job submission or execution. Correct the JCL in the job.</td>
</tr>
<tr>
<td>12</td>
<td>Error in the CTJXVER parameters or statements. Review the previous error messages in JOBLOG and SYSPRINT, and then correct the parameters or the control statements.</td>
</tr>
<tr>
<td>14</td>
<td>Member is already in use.</td>
</tr>
<tr>
<td>16 and above (except for 24, 30, and 112)</td>
<td>Internal error in the Control-M/JCL Verify product. Try again. If the problem reoccurs, contact BMC Customer Support.</td>
</tr>
<tr>
<td>20</td>
<td>Invalid input parameter. Internal error.</td>
</tr>
</tbody>
</table>
Example 1

In the following example, the CTJXVER edit macro is used to verify job ACCEP000. CTJXVER is invoked from the command line of the ACCEP000 member (located in the CTJP.TESTS.JCLS library) without options, as shown in Figure 65. Since the P option is not specified, the default parameters are used.
A screen capture of CTJXVER invoked from the command line of the ACCEP000 member is shown in Figure 66.

**NOTE**

Figure 66 shows that the macro JVER is invoked. This assumes that the CTJXVER edit macro was copied to the SYSPROC library as JVER.
After the CTJXVER macro command is submitted, the output of the verification process is displayed in the job member, as shown in Figure 67 and Figure 68. This output can be saved by copying and pasting, or by performing a screen print.

Figure 67  CTJXVER edit macro output– Example 1

The output is divided into two main sections. The header section, which begins just after the Top of Data line, displays the parameters used during the verification, a summary of the messages generated during the verification, and a list of the messages generated. The header section is following by the main body section that displays the job member line by line, with messages being displayed immediately after the lines to which they apply.

The first 6 columns in each row of the output header, which is usually reserved for indicating the line number of the job, indicates the type of message, a reference label, or other information as follows:
Example 1

- **MSG** - error message or label to job line in main body of output
- **------** - warning message or label to job line in main body of output
- **NOTE** - information message, label to job line in main body of output, or other information such as runtime parameters and job verification summary

In this example, the runtime parameters are: FA=Y FE=Y JES=Y ME=Y ML=I USER=N05. See Table 1 on page 30 and Table 2 on page 32 for explanations of these parameters.

In this example, the job verification summary indicates that there are no error messages, 3 warning messages, and 2 information messages. The first information message can be quickly located in the job using the label `.BAAA`. In the command line, enter the following command:

```
locate .BAAA
```

The display jumps to the following job line:

```
.BAAA // NOTIFY=&SYSUID
```

Note that the job line number (000002) was replaced by the label (.BAAA) during the verification process.

The next line that is displayed is the following information message:

```
=NOTE= IEF653I SUBSTITUTION JCL-0,YY,CLASS=A,MSGCLASS=X,NOTIFY=N05
```

The IEF prefix of the message indicates that this is an IBM generated message.

The advantage of the header becomes quickly apparent when verifying a lengthy job member. You can review the messages without scrolling through the entire job, and you can jump to the messages that interest you, using the labels.

In figure Figure 68, some of the color coding is apparent. For example, the job line numbers are blue and JCL job lines are displayed in green. The first 6 characters in each row, which includes message type indicators, and labels, are red if they were inserted during the verification process. Information messages and information inserted during the verification process are displayed in blue. Warning messages are displayed in white. Error messages are displayed in yellow.
Figure 68  CTJXVER edit macro output screen capture – Example 1

```
VIEW  CTJP.TESTS.JCL(ACCESSO) - 01.03  Columns 00001 00072
Command ===> ___________________________ Scroll ===> PAGE

***** __________________________________________ Top of Data ****************************
=NOTE=  RUNTIME PARAMETERS: FA=Y FE=Y JES=Y ME=Y CTM=N ML=I USER=N05
=NOTE=
=NOTE=  JOB VERIFICATION COMPLETED WITH
=NOTE=  0 ERRORS MESSAGES
=NOTE=  3 WARNINGS MESSAGES
=NOTE=  2 INFORMATION MESSAGES
=NOTE=
=NOTE=  ==> INFORMATION AT LABEL .BAAA
=NOTE=  IEFCS53I SUBSTITUTION JCL - 0,YY,CLASS=A,MSGCLASS=X,NOTIFY=N05
=NOTE=  ==> INFORMATION AT LABEL .BAAB
=NOTE=  INFORMATION: PGM IEFBR14 FOUND IN LINKLIST LIBRARY
====- ==> WARNING AT LABEL .BAAC
====- WARNING: NO ACCESS WAS GRANTED FOR DSN N18.ACCESS.NONE
====- ==> WARNING AT LABEL .BAAD
====- WARNING: UPDATE IS NOT ALLOWED FOR DSN N18.ACCESS.READ
====- ==> WARNING AT LABEL .BAAE
====- WARNING: DELETE IS NOT ALLOWED FOR DSN N18.ACCESS.READ
000001 //ACCESSO JOB 0,YY,CLASS=A,MSGCLASS=X,

.BAAA  //
  NOTIFY=&SYSUID
=NOTE=  IEFCS53I SUBSTITUTION JCL - 0,YY,CLASS=A,MSGCLASS=X,NOTIFY=N05
000003 /**  JCLLIB ORDER=IOAP.V700.PROCLIB
000004 //  JCLLIB ORDER=IOAP.V700.PROCLIB
.BAAB  //STEP1  EXEC PGM=IEFBR14
=NOTE=  INFORMATION: PGM IEFBR14 FOUND IN LINKLIST LIBRARY
=NOTE=  SYS1.LINKLIB
.BAAC  //ACCI00 DD DISP=SHR,DSN=N18.ACCESS.NONE
====- WARNING: NO ACCESS WAS GRANTED FOR DSN N18.ACCESS.NONE
000007 //ACCIYES DD DISP=SHR,DSN=N18.ACCESS.READ
.BAAD  //ACCI00 DD DISP=OLD,DSN=N18.ACCESS.READ
====- WARNING: UPDATE IS NOT ALLOWED FOR DSN N18.ACCESS.READ
.BAEE  //ACCI0000 DD DISP=(SHR,DELETE),DSN=N18.ACCESS.READ
====- WARNING: DELETE IS NOT ALLOWED FOR DSN N18.ACCESS.READ

********____________________________ Bottom of Data ****************************
```
Online job verification from IOA and Control-M interfaces

Control-M/JCL Verify can be invoked from the IOA Primary Option Menu for verifying JCL jobs. When Control-M is installed, Control-M/JCL Verify can be invoked from several Control-M user interface panels.

This chapter includes the following topics:

- IOA Primary Option Menu .......................................................... 154
- Control-M/JCL Verify defaults panel (JD screen) .......................... 156
- Control-M/JCL Verify Facility (JV screen) .................................... 158
- Job verification from Control-M interfaces ................................. 160
- Return Codes ............................................................................ 162
- Control-M/JCL Verify Site Standard Facility (JR screen) ............ 163
When Control-M/JCL Verify is installed, the IOA Primary Option Menu provides the following options for verifying JCL jobs, as shown Figure 69:

- **JD** - JCL Verify Defaults - for specifying verification parameter defaults
- **JV** - JCL Verification - for activating the verification process on job members
- **JR** - JCL Verify Rules - for specifying rules for site standard verification

*Figure 69  IOA Primary Option Menu showing Control-M/JCL Verify options*

If Control-M is installed, the JD and JV options are provided under the Control-M column, as shown in Figure 70.
Figure 70  Control-M/JCL Verify options with Control-M and Control-M/Restart

COMMANDS: X - EXIT, HELP, INFO, SET OR CHOOSE A MENU OPTION 16.03.35
Control-M/JCL Verify defaults panel (JD screen)

To open the Control-M/JCL Verify defaults panel (see Figure 71), enter JD in the option line in the IOA Primary Option Menu. Adjust the verification criteria settings or accept the current ones. The default settings are specified in the CTJPARM member.

Figure 71  Control-M/JCL Verify defaults panel

Table 40 describes the criteria for verifying the JCL members.
### Table 40  Control-M/JCL Verify verification criteria (part 1 of 2)

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JES SYNTAX</td>
<td>Whether to verify that the JES2 or JES3 statements are correct. Valid values are:</td>
</tr>
</tbody>
</table>
|                            |   ■ Y - Yes, always check  
|                            |   ■ N - No, never check  
|                            |   ■ D - as defined by the JESTMNT parameter in the CTJPARM member. Default |
| DATASET EXISTENCE          | Whether to verify file existence. Valid values are: |
|                            |   ■ Y - Yes, always check  
|                            |   ■ N - No, never check  
|                            |   ■ D - as defined by the DSNEXIST parameter in the CTJPARM member. Default |
| DATASET AUTHORIZATION      | Whether to verify file access privileges. Valid values are: |
|                            |   ■ Y - Yes, always check  
|                            |   ■ N - No, never check  
|                            |   ■ D - as defined by the DSNACCSS parameter in the CTJPARM member. Default |
| PROGRAM EXISTENCE          | Whether to verify that the load modules exist. Valid values are: |
|                            |   ■ Y - Yes, always check  
|                            |   ■ N - No, never check  
|                            |   ■ D - as defined by the PGMCHECK parameter in the CTJPARM member. Default |
| RESOLVE CONTROL-M AUTO-EDIT VARIABLE | Whether to resolve Control-M AutoEdit variables in the job. Valid values are: |
|                            |   ■ Y - Yes  
|                            |   ■ N - No  
|                            |   ■ D - as defined by the CTMVAR parameter in the CTJPARM member. Default |
| SITE STANDARDS RULES       | Whether to verify site standards. Valid values are: |
|                            |   ■ Y - Yes, always check  
|                            |   ■ N - No, never check  
|                            |   ■ D - as defined by the STDR parameter in the CTJPARM member. Default |
|                            | In the ENV field, specify which rules, in addition to the GENERAL rules, are to be used in the site standard verifications. The default value is GENERAL, indicating that only the GENERAL rules are to be used in the verifications |
The criteria are associated with each user and are saved in the following profile variables:

- SJVDJES (verify that the JES2 or JES3 statements are correct)
- SJVDDSEX (verify file existence)
- SJVDDSAU (verify file access privileges)
- SJVDPGEX (load modules existence test)
- SJVDUTL (the utility validation)
- SJVDMSGGL (minimum level of message severity to be issued)

### Control-M/JCL Verify Facility (JV screen)

**To verify job members**

1. Enter JV in the option line of the IOA Primary Option Menu.

   The Control-M/JCL Verify Facility - Entry Panel opens (see Figure 72).

2. In the LIBRARY field, enter the name of the library where the job members you want to verify are located.
In the MEMBER field, enter one of the options described in Table 41 to indicate which job members you want to verify, or leave the field blank for displaying the list of members, as shown in Figure 73.

**Table 41** CTJVER options for MEMBER field

<table>
<thead>
<tr>
<th>Enter a...</th>
<th>To perform...</th>
</tr>
</thead>
<tbody>
<tr>
<td>blank</td>
<td>to open a list of all the job members in the specified library (Start the job verification by entering E or S in the OPT column next to the job member name. Note: E or S produce the same results.)</td>
</tr>
<tr>
<td>*</td>
<td>to verify all the job members in the specified library</td>
</tr>
<tr>
<td>mask</td>
<td>to specify a mask for verifying a group of job members</td>
</tr>
<tr>
<td>name</td>
<td>to specify the name of the job member to be verified</td>
</tr>
</tbody>
</table>
4 Type S or E in the OPT column next to the name of the member that you want to verify.

5 Press ENTER.

A verification report is displayed, consisting of the JCL statements with the Control-M/JCL Verify informative, warning, and error report messages appearing where applicable.

Job verification from Control-M interfaces

Verifies all the JCLs pointed from or in-streamed in all the job definitions in the table. The verifications are according to the defaults specified in the JD option. The output of the verifications is displayed in view mode. Table 42 indicates which commands and options are available for invoking job verification from the various Control-M interfaces.
### Table 42  Options and commands for verification from Control-M interfaces

<table>
<thead>
<tr>
<th>Screen #/Description</th>
<th>Interface</th>
<th>Pop-up confirmation window</th>
<th>Option</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Table List</td>
<td></td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>Control-M Job</td>
<td></td>
<td></td>
<td>Order/Force</td>
<td>Y (VERIFY JCL)</td>
</tr>
<tr>
<td>Scheduling Definition</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Job List</td>
<td></td>
<td>E</td>
<td>Order/Force</td>
<td>Y (VERIFY JCL)</td>
</tr>
<tr>
<td>Job Scheduling</td>
<td></td>
<td></td>
<td></td>
<td>JVER</td>
</tr>
<tr>
<td>Definition</td>
<td></td>
<td></td>
<td></td>
<td>CTJXVER (global name of edit macro)</td>
</tr>
<tr>
<td>View JCL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Job List</td>
<td></td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>Control-M Active</td>
<td></td>
<td></td>
<td>Rerun/Restart</td>
<td>Y (Verify JCL)</td>
</tr>
<tr>
<td>Environment Display</td>
<td></td>
<td></td>
<td></td>
<td>JVER</td>
</tr>
<tr>
<td>Job Save</td>
<td></td>
<td></td>
<td></td>
<td>CTJXVER (global name of edit macro)</td>
</tr>
<tr>
<td>View JCL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**NOTE**

If MULTJOBS=Y is specified in the CTMPARM member, only the first job will be submitted from the JCL member by Control-M. Therefore, if the JCL member to be verified contains more than one job, only the first job will be verified.
## Return Codes

### Table 43  Return Codes (part 1 of 2)

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Operation performed successfully.</td>
</tr>
<tr>
<td>4</td>
<td>Control-M/JCL Verify discovered that the verified resource is not in the required status at the time of the verification. For example, an input dataset does not exist. Review the status and decide if there is a need to make a change in the job.</td>
</tr>
<tr>
<td>8</td>
<td>Control-M/JCL Verify discovered a problem in the job that will cause it to fail with a JCL error or to be rejected by JES2 or JES3, either during job submission or execution. Correct the JCL in the job.</td>
</tr>
<tr>
<td>12</td>
<td>Error in the CTJXVER parameters or statements. Review the previous error messages in JOBLOG and SYSPRINT, and then correct the parameters or the control statements.</td>
</tr>
<tr>
<td>16 and above (except for 24, 30, and 112)</td>
<td>Internal error in the Control-M/JCL Verify product. Try again. If the problem reoccurs, contact BMC Customer Support.</td>
</tr>
<tr>
<td>20</td>
<td>Invalid input parameter. Internal error.</td>
</tr>
<tr>
<td>24</td>
<td>Under ISPF: Control-M/JCL Verify is not APF-authorized.</td>
</tr>
<tr>
<td>30</td>
<td>GETMAIN failed. The user needs to increase the TSO region.</td>
</tr>
<tr>
<td>32</td>
<td>GETMAIN error during initialization. Increase user region size.</td>
</tr>
<tr>
<td>34</td>
<td>IOA INIT failed. Review the TSO user’s JESLOG for additional messages, and contact BMC Customer Support.</td>
</tr>
<tr>
<td>35</td>
<td>CTJAPI INIT failed. Review the TSO user’s JESLOG for additional messages, and contact BMC Customer Support.</td>
</tr>
<tr>
<td>36</td>
<td>GETMAIN error during initialization. Increase user region size.</td>
</tr>
<tr>
<td>38</td>
<td>A parameter scan error. One or more input parameters are not valid. Perform CTJXVER P to correct the values. If the problem reoccurs, contact BMC Customer Support.</td>
</tr>
<tr>
<td>42</td>
<td>Error on IKJCT441. Internal error, check TSO user’s JESLOG for additional messages, and contact BMC Customer Support.</td>
</tr>
<tr>
<td>46</td>
<td>No data returned from the Verify function (empty buffer). Internal error, check TSO user’s JESLOG for additional messages, and contact BMC Customer Support.</td>
</tr>
<tr>
<td>50</td>
<td>An abend occurred returning from recovery routine MLIJ10034T. Review TSO user’s JESLOG for additional messages, and call BMC Customer Support.</td>
</tr>
<tr>
<td>54</td>
<td>Control-M/JCL Verify password is invalid or expired.</td>
</tr>
</tbody>
</table>
Table 43  Return Codes (part 2 of 2)

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>58</td>
<td>The specified member or library does not exist.</td>
</tr>
<tr>
<td>112</td>
<td>Under all environments except ISPF: Control-M/JCL Verify is not APF-authorized.</td>
</tr>
</tbody>
</table>

Control-M/JCL Verify Site Standard Facility (JR screen)

For information about the Control-M/JCL Verify Site Standard Facility, see Chapter 3, “Site standards verification.”
The Control-M/JCL Verify Application Program Interface (API)

Overview

The Control-M/JCL Verify Application Program Interface (CTJAPI) is an open interface between the application environment and Control-M/JCL Verify. CTJAPI enables your application program to interface with Control-M/JCL Verify so that you can access services and extract data from Control-M/JCL Verify into your own programs.
The following CTJAPI functions are available:

- Initialize the Control-M/JCL Verify environment
- Perform JCL verification
- Perform cleanup when terminating the Control-M/JCL Verify environment

CTJAPI macro and keywords

The CTJAPI macro is the interface service API for Control-M/JCL Verify. It allows the user to define the communication block and call CTJAPI to perform the requests.

The macro automatically changes keywords that must be in upper case. For example, jobname is changed to JOBNAME.

An example of the CTJAPI macro is shown in Figure 74 and the macro keywords are described in Table 44.

Figure 74  CTJAPI macro example

| INIT | CTJAPI INIT,RAREA=APIJ |
| SCAN | CTJAPI SCAN,RAREA=APIJ,MCT=(R12),USER=USRID |
| END  | CTJAPI END, RAREA=APIJ |
|      | TITLE 'CTJAPI: CTJ COMMUNICATION AREA' |
| APIJ | CTJAPI APID |

Refer to the CTJAPIA member in the IOA.SAMPLE library for an example of a program that uses the macro.
### Table 44  Macro keywords (part 1 of 4)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Func</strong></td>
<td>Positional parameter that describes the required service. There are no defaults. Mandatory. Valid values:</td>
</tr>
<tr>
<td></td>
<td>■ INIT - Initializes the Control-M/JCL Verify environment. See notes above.</td>
</tr>
<tr>
<td></td>
<td>— When Func is INIT, the block is reformatted and the fields, if not set, will contain X'00'.</td>
</tr>
<tr>
<td></td>
<td>— When a keyword is not used, the macro does not change the related field.</td>
</tr>
<tr>
<td></td>
<td>■ SCAN - performs JCL validation</td>
</tr>
<tr>
<td></td>
<td>■ END - performs cleanup when terminating the interface. See note in JMSG</td>
</tr>
<tr>
<td></td>
<td>■ APID - generates the API Communication Area DSECT</td>
</tr>
<tr>
<td><strong>DSECT</strong></td>
<td>Defines the CTJAPI block as a DSECT. Valid values:</td>
</tr>
<tr>
<td></td>
<td>■ Y - Yes</td>
</tr>
<tr>
<td></td>
<td>■ N - No</td>
</tr>
<tr>
<td><strong>FA</strong></td>
<td>Whether to verify file access privileges. Valid values:</td>
</tr>
<tr>
<td></td>
<td>■ Y - Yes</td>
</tr>
<tr>
<td></td>
<td>■ N - No</td>
</tr>
<tr>
<td></td>
<td>■ D - as defined by the DSNACCSS parameter in the CTJPARM member. Default</td>
</tr>
<tr>
<td><strong>FE</strong></td>
<td>Whether to verify file existence. The file is verified whether it is cataloged and whether it is in the volume as specified in the DD statement or in the catalog. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>■ Y - Yes</td>
</tr>
<tr>
<td></td>
<td>■ N - No</td>
</tr>
<tr>
<td></td>
<td>■ D - as defined by the DSNEXIST parameter in the CTJPARM member. Default</td>
</tr>
<tr>
<td><strong>CTM</strong></td>
<td>Whether to resolve Control-M AutoEdit variables in the job. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>■ Y - Yes</td>
</tr>
<tr>
<td></td>
<td>■ N - No</td>
</tr>
<tr>
<td></td>
<td>■ D - as defined by the CTMVARs parameter in the CTJPARM member. Default</td>
</tr>
</tbody>
</table>
### Table 44  Macro keywords (part 2 of 4)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
</table>
| STDR    | Whether to use the Control-M/JCL Verify rules to verify site standards. Valid values:  
  - Y - Yes  
  - N - No  
  - D - as defined by the STDR parameter in the CTJ Parm member. Default |
| ENV     | Whether to use, in addition to the GENERAL rules, rules that are defined for specific environments for the site standard verifications. Valid values are:  
  - Y - Yes  
  - N - No  
  - D - empty - only GENERAL rules are used. Default |
| JCL#    | Number of JCL statements passed to CTJ API |
| JCL@    | Address of buffer containing the JOB to be verified.  
  Note: Only one job can be verified. In case the buffer contains more than one job only the first job will be handled |
| JES     | Whether to verify that the JES2 or JES3 statements are correct. Valid values are:  
  - Y - Yes  
  - N - No  
  - D - as defined by the JESTMTNT parameter in the CTJ Parm member. Default |
| JMSG    | Get the Control-M/JCL Verify messages for the job from the message buffer. Valid values:  
  - Y - Yes  
  - N - No |

After the scan is completed the following fields are set:  
- JAPIMSG@ - address of message buffer. The address of the buffer is above the BAR, meaning that the address is to a double-word address area.  
- JAPIMSG# - number of messages in buffer  
- JAPIMSGSZ - buffer size in a double-word area  

Note: The calling program is responsible for releasing the message buffer.
## Table 44 Macro keywords (part 3 of 4)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>JOB</strong></td>
<td>The name of the Control-M job scheduling definition to be verified. Control-M must be D or Y. Control-M must be installed with ICTCTM=Y.</td>
</tr>
<tr>
<td><strong>LIB</strong></td>
<td>Library containing the JCL members to be verified or Control-M Schedule job definitions library, depending on the value of the TYPE parameter. Mandatory.</td>
</tr>
<tr>
<td><strong>MEMBER</strong></td>
<td>The name of the member, or masking of the members, to be verified or the Control-M schedule job definitions member, depending on the value of the TYPE parameter. If * or null, all members in the library will be verified.</td>
</tr>
</tbody>
</table>
| **MCT** | MCT address. Mandatory. Valid values:  
- (R12) - Default  
- =0 - MCT field is set to 0 and the API module will initialize the IOA environment  
- Null - the MCT field will not be changed  
- address of MCT block  
- (Rx) - address of MCT block in any register 2 to 12 |
| **ME** | Module existence. The module in EXEC PGM=module_name is existence is verified.  
- If the PGM is not found, a warning message will be issued.  
- If the PGM is found, an information message that contains from where the PGM (DD name or LINKLIST and library name)  
Valid values:  
- Y - Yes  
- N - No  
- D - as defined by the PGMCHECK parameter in the CTJPARM member. Default |
| **ML** | The minimum level of message severity to be issued. Valid values are:  
- I – Information – all messages are issued.  
- W – warning messages and errors are issued.  
- E – Only error messages are issued  
- D - as defined by the MSGLEVEL parameter in the CTJPARM member. Default  
Note: JMSG=Y all messages will be available |
The application program can use the verification report information if the report records are retrieved from the message buffer. The application program requests the retrieval with the JMSG API parameter set to Y. The messages in Table 45 are used to indicate the record types in the message buffer.

After the scan is completed, the following fields in the CTJAPI block are set:

### Table 44 Macro keywords (part 4 of 4)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
</table>
| RAREA   | Label of CTJAPI communication block. Mandatory. Valid values:  
  - APID - the default name when creating a CTJAPI block using: CTJAPI APID |
| USER    | The user ID to be used in the file access privileges (FA) verifications. Valid values are:  
  - user_ID  
  - *DEFAULT - use the default user ID (either the user invoking the CTJVER utility, or in the case of a Control-M job definition, the Control-M owner - as specified in the OWNER field in the job definition)  
  
  The USER is used in all the statements that follows it until a new USER is specified). If the USER statement is omitted, then USER=*DEFAULT.  
  
  Note: If the JOB statement contains USER=, then this user is used. |
| UT      | Whether to verify the syntax of the IEBGENER, IEBCOPY, and SORT IBM utilities. Valid values are:  
  - Y - Yes  
  - N - No  
  - D - as defined by the SUPUTIL parameter in the CTJPARM member. Default  
  
  Note: UT=Y must be used with ME=Y, FE=Y, and FA=Y. |
| ODATE   | The Control-M Order Date, which the user can change. If it is left blank, the default is the current date. The format of it is: YYMMDD |
■ JAPIMSG@ - address of message buffer. The address of the buffer is above the BAR, meaning that the address is to a double-word address area. The program that is accessing the message buffer must run in AMODE 64.

■ JAPIMSG# - number of messages in buffer

■ JAPIMSGSZ - buffer size in a double-word area

The calling program is responsible for releasing the message buffer.

Table 45 Internal messages

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTJ100I</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>CTJ101I</td>
<td>Indicates an original JCL statement or comment in the verified member.</td>
</tr>
<tr>
<td>CTJ102I</td>
<td>Indicates a JCL statement or comment that was added to expand the JCL procedure in the verified member. The source of the addition is indicated by one of the following symbols, which starts the JCL text, as follows:</td>
</tr>
<tr>
<td></td>
<td>■ xx - JCL statement from a JCL PROCEDURE or INCLUDE member</td>
</tr>
<tr>
<td></td>
<td>■ xx* - JCL comment from a JCL PROCEDURE or INCLUDE member</td>
</tr>
<tr>
<td></td>
<td>■ ++ - JCL statement from an instream JCL PROCEDURE</td>
</tr>
<tr>
<td></td>
<td>■ ++* - JCL comment from an instream JCL PROCEDURE</td>
</tr>
<tr>
<td>CTJ103I</td>
<td>Indicates an IBM message that normally appears in the JESJCL or JESYSMSG when the JOB is submitted and executed.</td>
</tr>
<tr>
<td>CTJ104I</td>
<td>Indicates a SYSIN record.</td>
</tr>
<tr>
<td>CTJ105I</td>
<td>Indicates a “//SYSIN DD” statement, which Control-M/JCL Verify generates when simulating JES2 or JES3. When the job is submitted, an asterisk is added as follows: //SYSIN DD *</td>
</tr>
<tr>
<td>CTJ106I</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>CTJ107I</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>CTJ108I</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>CTJ109I</td>
<td>A null (//) JCL statement indicates the end of a job.</td>
</tr>
</tbody>
</table>
CTJINIT utility

The CTJINIT utility boosts the performance of the JCL verifications by loading basic site related information into shared objects.

This chapter includes the following topics:

Overview ................................................................. 174
Parameters ............................................................. 175
Activating the Utility ................................................. 176
Batch JCL ................................................................. 176
Operator commands .................................................. 177
Return Codes .......................................................... 178
Overview

The CTJINIT utility boosts the performance of the JCL verifications by loading basic site related information into a shared object allocated in “above the bar” memory. The loaded information is later accessed by all subsequent invocations of Control-M/JCL Verify until the shared object is removed. The utility is optional and the user can choose whether to use it to gain the performance boost.

Each invocation of Control-M/JCL Verify scans the shared object for the information and if it exists will use it. If the shared object does not exist (either it was not loaded or already removed) then each invocation gathers its own copy of the information with transparency to the user (although a significant increase in the response time might be noticed).

The following message is issued when the shared object is not available:

CTJ202I Shared INFO not found. Run CTJINIT to improve verification response time

The information loaded into the shared object includes:

- Programs found in the system’s LINKLIST and LPA
- Mounted disk volumes
- Available device units
- JES Procedures libraries
- JES input and output classes
- JES destinations
- Printers and punches
- NJE nodes
- FCB images
- Additional JES2 or JES3 information that is required to check JCL and JES statements
- Site standard rules

The loaded data is static and is not refreshed automatically. For example, if a new module is added to the LPA after the shared object was loaded, it will not be recognized by the subsequent verifications unless the data is refreshed. Therefore, although the nature of the data kept in the shared object is static, it is recommended that the data is refreshed on a periodic basis.
Table 46 describes the parameters that the CTJINIT utility receives from the JCL procedure:

### Table 46  CTJINIT functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| REFRESH  | The utility loads the information and creates a new shared object allocated in “above the bar” memory.  
If a shared object already exists (as a result of a preceding REFRESH) and is not in use by a verification process, the shared object is removed, and a new shared object is created.  
If a shared object already exists and is in use by a verification process, a new shared object is created. The memory of the old shared object will not be released until one of the next REFRESH or REMOVE executions finds it unused. (Note that this is not considered a problem and therefore no error message will be issued).  
If no parameters are supplied to CTJINIT, the default value is set to REFRESH. |
| REMOVE   | Removes the shared object (if it exists), provided it is not currently in use by a running verification process.  
An error message is issued if the shared object is not found or if it is found currently in use and therefore cannot be removed.  
When the shared object is in use, the user can try using REMOVE again after a few seconds or minutes. Alternatively, the user can use REMOVEF, as described below. |
| REMOVEF  | Forcefully removes the shared object (if it exists) even if it is currently in use by a running verification process. An error message is issued if the shared object is not found.  
**Warning:** Any running verification process that uses this shared object will abend.  
After the running REMOVEF, stop and then restart the IOAOMON monitor, if it is already started. |
The SCOPE parameter, used with the REFRESH, REMOVE, and REMOVEF functions, determines which data is loaded or removed. When SCOPE is set to ALL, the loading or removing function is applied to all Control-M/JCL Verify tables and the rules listed in the CTJRULE member (located in the olprefj.PARM library). When SCOPE is set to RULES, the loading or removing function is applied to only to the rules listed in the CTJRULE member.

Activating the Utility

It is advisable to run the CTJINIT utility automatically in REFRESH mode as follows:

- Right after the system is initialized (IPLed) and JES2 or JES3 is active
- On a periodic basis (for example, on an hourly basis)

Batch JCL

The following is a sample batch JCL used to invoke CTJINIT:

```jcl
// .... JOB ....
//INIT EXEC CTJINIT,FUNCTION=REFRESH|REMOVE|REMOVEF,SCOPE=ALL|RULES
/*
//CTJINIT JOB 0,YY,CLASS=A,MSGCLASS=X, NOTIFY=&SYSUID
// JCLLIB ORDER=IOAP.V800.PROCLIB
// INCLUDE MEMBER=IOASET
//*------------------------------------------------------------------*
//**                                                                 *
//**   PARAMETERS TO CTINIT                                          *
//**     FUNCTION=REFRESH - loads the shared pools into shared memory* *
//**     FUNCTION=REMOVE - removes all unused shared-pools             *
//**     FUNCTION=REMOVEF - removes all shared-pools regardless of     *
//**                        whether they are used or not                 *
//**     SCOPE=ALL|RULES - determines which data is loaded or removed* *
//*------------------------------------------------------------------*
//INIT EXEC CTJINIT,FUNCTION=REFRESH
/*
```

Sample job CTJINITJ is located in the IOA.JCL library.

The following job is an example in which FUNCTION is set to REFRESH:
The following is a sample output using FUNCTION=REFRESH:

```
19.01.53 CTJ200I CTJINIT is Starting Level 8.0.02 APAR: IJ10085
19.02.05 CTJ20BI Control-M/JCL Verify Shared Pool was created successfully
19.02.05 CTJ20AI CTJINIT Function REFRESH  ended - RC=0000
```

The following job is an example in which FUNCTION is set to REMOVE:

```
//CTJINIT  JOB 0,YY,CLASS=A,MSGCLASS=X,                    
         NOTIFY=&SYSUID
// JCLLIB ORDER=IOAP.V800.PROCLIB
// INCLUDE MEMBER=IOASET
//**----------------------------------------------------------------------*
//** PARAMETERS TO CTINIT                                              *
//** FUNCTION=REFRESH - loads the shared pools into shared memory    *
//**                to be available for searching routines             *
//** FUNCTION=REMOVE  - removes all unused shared-pools              *
//** FUNCTION=REMOVEF  - removes all shared-pools regardless of      *
//**       whether they are used or not                              *
//** SCOPE=ALL|RULES - determines which data is loaded or removed    *
//**----------------------------------------------------------------------*

//CTJINIT  EXEC  CTJINIT,FUNCTION=REMOVE
//**----------------------------------------------------------------------*
```

The following is a sample output using FUNCTION= REMOVE:

- Shared Pool is not use by any user

```
19.07.32 CTJ200I CTJINIT is Starting Level 8.0.02 APAR: IJ10085
19.07.32 CTJ20AI CTJINIT Function REMOVE  ended - RC=0000
```

- Shared Pool is in use by at least one user

```
19.03.06 CTJ200I CTJINIT is Starting Level 8.0.02 APAR: IJ10085
19.03.06 CTJ20AI Shared Pool is in use and was not removed
19.03.06 CTJ20AI CTJINIT Function REMOVE  ended - RC=0004
```

Operator commands

The operator can perform the various CTJINIT functions by using the following commands:

**To load the site standard rules**

- `S ctjJREF,SCOPE=RULES`
■ $ctjJREF,SCOPE=ALL

To remove the site standard rules

■ $ctjJREM,SCOPE=RULES
■ $ctjJREM,SCOPE=ALL

To forcefully remove the site standard rules

■ $ctjJREMF,SCOPE=RULES
■ $ctjJREMF,SCOPE=ALL

where $ctj$ are the first three characters of the Control-M/JCL Verify JCL procedures defined in PROCPRFJ.

The members must be in the system Started Task jobs library.

Return Codes

Table 47  CTJINIT Return Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Operation performed successfully.</td>
</tr>
</tbody>
</table>
| 4    | ■ FUNC=REMOVE: at least one shared object is in use  
      ■ FUNC=REMOVE/REMOVEL: shared object does not exist |
| 12   | Parameter error in PARM. |
| 16   | CTJINIT FUNC=REMOVE was performed before CTJINIT  
      FUNC=REFRESH was performed to create the shared object, and  
      therefore there is nothing to remove. |
| 20   | Internal error - one of the searching routines failed. Check for  
      previous messages in SYSPRINT or JOBLOG, and contact BMC  
      Customer Support. |
| 24 and above | Internal error in the Control-M/JCL Verify product. Try again. If the problem reoccurs, contact BMC Customer Support. |
| 36   | Internal error - Failed to load one of the searching routines. Check  
      the region size and the STEPLIB statement, and try again. If the  
      problem reoccurs, contact BMC Customer Support. |
| 44   | Control-M/JCL Verify is not APF-authorized. |
Messages

CTJ000I  Control-M/JCL Verify is starting. Level *apar_id*

*Explanation:* The message is issued when Control-M/JCL Verify starts the first time.

*User Response:* No action is required.

CTJ001I  Control-M/JCL Verify is ended RC=*return_code*

*Explanation:* The message is issued when Control-M/JCL Verify completes the validation processing. RC is the highest return code that occurs during the validation processing.

*User Response:* If RC is not zero, it means that at least one validation has failed. Review the warning and error messages and handle them appropriately.

CTJ002I  Start verifying MEMBER *member_name* DSNAME=*dataset_name*

*Explanation:* The message is issued when Control-M/JCL Verify starts processing the JCL of the *member_name* member from the *dataset_name* dataset.

*User Response:* No action is required.

CTJ003I  Processing ended RC=*return_code* REASON *reason_code* MEMBER *member_name* DSNAME *dataset_name*

*Explanation:* The message is issued when Control-M/JCL Verify finished processing the JCL of the *member_name* member from the *dataset_name* dataset. RC is the highest return code that occurs during the validation processing. REASON is the reason code of the highest return code.

*User Response:* If RC is not zero, it means that at least one validation has failed. Review the warning and error messages and handle them appropriately.

CTJ004I  Verified *number* members from *lib_name*

*Explanation:* The message is issued when Control-M/JCL Verify finished verifying the *lib_name* library, which contains *number* members.
CTJ005E  MEMBER member_name not found in lib_name

Explanation: The JCL verification failed because the member_name member does not
exist in the lib_name library. Return code is 12. Reason code is 00080014.
User Response: Correct the member or library name and re-submit the Control-
M/JCL Verify job.

CTJ006E  Could not find LIBRARY=lib_name

Explanation: The JCL verification failed because the lib_name library cannot be found.
Return code is 12. Reason code is 00010014.
User Response: Check the library name. If necessary, change or correct the library
name.

CTJ007E  Empty MEMBER member_name in library lib_name

Explanation: The JCL verification failed because the member_name member, in the
lib_name library, is empty. Return code is 12. Reason code is 0.
User Response: Check the member and library names. If necessary, change or correct
the appropriate name.

CTJ008I  Start verifying JOB job_name MEMBER member_name

Explanation: The message is issued when Control-M/JCL Verify starts processing a
JCL JOB from the member_name member.
User Response: No action is required.

CTJ009I  Processing ended RC=return_code for JOB job_name MEMBER member_name

Explanation: The message is issued when Control-M/JCL Verify completed
processing a JCL JOB from the member_name member. RC is the highest return code
occurring during the validation process.
User Response: If RC is not zero, it means that at least one validation failed. Review
the warning and error messages and handle them appropriately.

CTJ00AE Control-M/JCL Verify is not APF-authorized

Explanation: Control-M/JCL Verify must be executed in an APF-authorized
environment. The Control-M/JCL Verify initialization process found that the
environment is not APF-authorized.
User Response: Verify that the JOBLIB or STEPLIB are all defined as APF- authorized.
Under TSO, validate that the ISPF Edit Macro is defined as an Authorized command.
CTJ00BE  Reading library's directory using mask mask failed DSNAME=dataset_name

Explanation: The Control-M/JCL Verify initialization process failed to retrieve any member from the specified library.

User Response: Check that the mask masking name is valid and results in at least one member name.

CTJ00CI  Runtime parameters: FA=fa FE=fe JES=jes ME=me ML=ml CTM=ctm USER=user

Explanation: Control-M/JCL Verify displays the values of the control parameters before scanning is performed. The message is issued whenever one of the parameters is changed.

User Response: No action is required.

CTJ00II  Job ID jobid in Jobs Dependency Flow.

Explanation: The message is issued when Control-M/JCL Verify recognizes the job ID of each of the jobs in the flow. To recognizes the job IDs, the Jobs Dependency flag must be set to “On.”

User Response: No action is required.

CTJ00SE  jcl_statement_number ERROR: text

Explanation: The user specified the DO MSG action parameter in the DO section of the Control-M/JCL Verify Rule definition. As part of the DO MSG definition, the user specified "E" (for an error type message) and typed in text for the actual message.

User Response: No action is required.

CTJ00SI  jcl_statement_number INFORMATION: text

Explanation: The user specified the DO MSG action parameter in the DO section of the Control-M/JCL Verify Rule definition. As part of the DO MSG definition, the user specified "I" (for an information type message) and typed in text for the actual message.

User Response: No action is required.

CTJ00SW  jcl_statement_number WARNING: text

Explanation: The user specified the DO MSG action parameter in the DO section of the Control-M/JCL Verify Rule definition. As part of the DO MSG definition, the user specified "W" (for a warning type message) and typed in text for the actual message.

User Response: No action is required.
CTJ00TI  
\[ jcl\_statement\_number \] REXX-INFO: message text

Explanation: The message text is an information message issued by a DOMSG from an EXEC.

User Response: No action is required.

CTJ00TW  
\[ jcl\_statement\_number \] REXX-WARN: message text

Explanation: The message text is a warning message issued by a DOMSG from an EXEC.

User Response: No action is required.

CTJ00TE  
\[ jcl\_statement\_number \] REXX-ERR: message text

Explanation: The message text is an error message issued by a DOMSG from an EXEC.

User Response: No action is required.

CTJ105I  //SYSIN DD * GENERATED BY CTJVER

Explanation: The message is issued if Control-M/JCL Verify finds instream data (a.k.a. SYSIN), which is not preceded by a "DD *" statement. Control-M/JCL Verify simulates the JES2 and JES3 behaviors and generates the //SYSIN DD statement.

User Response: No action is required.

CTJ200I CTJINIT is Starting Level level_id APAR:apar_id

Explanation: The message is issued when CTJINIT starts.

User Response: No action is required.

CTJ201E ERROR: string tables were not initiated

Explanation: The message is issued when Control-M/JCL Verify detects that the shared tables were not initiated.

User Response: Run the CTJINIT utility with FUNC=REFRESH.

CTJ202I Shared INFO not found. Run CTJINIT to improve verification response time

Explanation: The message is issued when Control-M/JCL Verify detects that the shared tables were not initiated.

User Response: Run the CTJINIT utility with FUNC=REFRESH.

CTJ203E ERROR: Storage not available Return Code:return_code, Reason:reason_code

Explanation: The message is issued when the application cannot allocate memory objects. The return and reason codes indicates those returned by the IARV64 system service.
User Response: Check the IARV64 return and reason codes in the *IBM MVS Authorized Assembler Services Reference*.

**CTJ204E**  
**ERROR: JES sub-system is not available**

*Explanation:* The message is issued when the application cannot locate the JES Sub-System in the sub-system table.

*User Response:* Consult with your z/OS system administrator.

**CTJ205E**  
**ERROR: string tables cannot be accessed. Return Code: return_code**

*Explanation:* The message is issued when the application cannot access the JES2 or JES3 data areas due to ALESERV failure.

*User Response:* Consult with your z/OS system administrator.

**CTJ206E**  
**ERROR: Sub-System internal error**

*Explanation:* The message is issued when the JES data area does not contain the expected information.

*User Response:* This is an internal error. Retry the process and if the problem reoccurs, call BMC Customer Support.

**CTJ207E**  
**ERROR: Sub-System failure. Return Codes: return_code - reason_code**

*Explanation:* The message is issued when Control-M/JCL Verify calls the JES subsystem and the JES response is not 0.

*User Response:* Verify that the JES2 or JES3 is active. If the JES2 or JES3 is active, retry the process again. If the problem reoccurs, check the IEFSSREQ return and reason codes in the *IBM MVS Authorized Assembler Services Reference*.

**CTJ208E**  
**ERROR: CTJINIT input parameters invalid. RC=return_code**

*Explanation:* The message is issued because the input parameters supplied to CTJINIT in PARM are invalid.

*User Response:* Correct the parameter and re-submit the CTJINIT utility.

**CTJ209E**  
**ERROR: STEPLIB dataset is not a library, DSN=dataset_name**

*Explanation:* The message is issued when Control-M/JCL Verify accepts a file as a LINK-LIST library, but it is not a library.

*User Response:* Correct the JCL statement and rerun the job verification.

**CTJ20A1**  
**CTJINIT Function function Scope string ended - RC=return_code**

*Explanation:* The message is issued when CTJINIT terminates.

*User Response:* If RC is greater than 0, check for previous messages.
<table>
<thead>
<tr>
<th>CTJ20BI</th>
<th>Control-M JCL Verify Data Pool was created successfully</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explanation:</td>
<td>The message is issued after the searching routine loads its table successfully.</td>
</tr>
<tr>
<td>User Response:</td>
<td>No action is required.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CTJ20CE</th>
<th>Data Pools were not created</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explanation:</td>
<td>The message is issued if the searching routine is unsuccessful in loading its table.</td>
</tr>
<tr>
<td>User Response:</td>
<td>Check for previous messages and make the necessary corrections.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CTJ20DW</th>
<th>Data Pool is in use and was not removed. SCOPE=string</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explanation:</td>
<td>The message is issued while removing old data pools and one of the data pools is in use.</td>
</tr>
<tr>
<td>User Response:</td>
<td>Re-submit the CTJINIT utility with FUNC=REMOVE, provided that no user is currently performing a JCL verification.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CTJ20EW</th>
<th>JES search table string not recognized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explanation:</td>
<td>The message is issued while removing old shared pools and one of the shared pools is in use.</td>
</tr>
<tr>
<td>User Response:</td>
<td>Re-submit the CTJINIT utility with FUNC=REMOVE, provided that no user is currently performing a JCL verification.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CTJ20FE</th>
<th>Control-M/JCL Verify system anchor was not created. RC=return_code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explanation:</td>
<td>The message is issued if CTJINIT FUNC=REMOVE is submitted before CTJINIT FUNC=REFRESH was submitted to create the shared tables, and therefore there is nothing to remove.</td>
</tr>
<tr>
<td>User Response:</td>
<td>No action is required.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CTJ20GE</th>
<th>ERROR: string library info cannot be obtained for library_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explanation:</td>
<td>The message is issued when the CTJSHRM tries to assign a new entry in SMC, and there is no available entry.</td>
</tr>
<tr>
<td>User Response:</td>
<td>Run CTJINIT with FUNCTION=REMOVE. If it does not help, run CTJINIT with FUNCTION=REMOVEF and then FUNCTION=REFRESH.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CTJ20HE</th>
<th>Number of tasks exceeds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explanation:</td>
<td>The message is issued when the CTJSHRM tries to assign a new entry in SMC, and there is no available entry.</td>
</tr>
<tr>
<td>User Response:</td>
<td>Consult with your z/OS system administrator.</td>
</tr>
</tbody>
</table>
CTJ20IE  internal error: Shared Memory Object was not found

Explanation: The message is issued when supposed pointer of SMC is null.
User Response: Consult with your z/OS system administrator.

CTJ20JE  CTJSHRM - Not Enough Memory. STORAGE RC=return_code

Explanation: The message is issued when the CTJSHRM tries to assign a new entry in SMC, and there is no available entry.
User Response: Consult with your z/OS system administrator.

CTJ20KE  PDS pds_name cannot be accessed due to authorization issue

Explanation: The message is issued when trying to read STEPLIB/JOBLIB/ LINKLIST library and the OPEN INPUT got S913 (Access not permitted).
User Response: Grant READ permission to the mentioned PDS.

CTJ20LW  WARNING: Library volume volume_id cannot be accessed DSN=dataset_name

Explanation: The message is issued when a VOLSER of STEPLIB/JOBLIB/ LINKLIST library is not mounted.
User Response: Check the volume existence mentioned in the message.

CTJ20ME  Syntax error during CTJPRC parsing. Line:line_number

Explanation: The message is issued during parsing of the procedure libraries according to the rules defined in the CTJPRC member.
User Response: Check syntax of CTJPRC and recode it according to the syntax rules.

CTJ20NE  Unrecognized System Variable Name:variable_name

Explanation: The message is issued during parsing of the procedure libraries which contain a variable that is not among the system variable names.
User Response: Check the name preceded by '&'. Recode it, or remove.

CTJ20OI  Procedure Library was not Found. Library:library

Explanation: The message is issued when trying to LOCATE a procedure library in the catalog fails.
User Response: Check the PROCLIBs in CTJPRC for its existence.

CTJ20PE  No JES Procedure libraries provided with string DD name

Explanation: The message is issued when trying to LOCATE a procedure library in the catalog fails.
**User Response:** Create CTJPRC, according the rules.

**CTJ20QI** IN\_FORMATION: LLA Library is not authorized for reading

*Explanation:* The message is issued when security check for reading failed.

*User Response:* Grant READ permission to all LINK\-LIST libraries.

**CTJ20RE** CTJINIT utility is already running. Execution aborted.

*Explanation:* The message is issued when an attempt is made to run two instances of CTJINIT in the same system.

*User Response:* Wait until one of the instances of CTJINIT terminates.

**CTJ20SE** Rule List member *member\_name* is empty

*Explanation:* The message is issued when the member containing the list of rules is empty.

*User Response:* Create the rule list.

**CTJ301E** ERROR: The /*PRIORITY statement must precede the JOB statement.

*Explanation:* The message is issued when Control-M/JCL Verify detects that a /*PRIORITY statement appears after a JOB statement.

*User Response:* Move the /*PRIORITY statement to appear before the JOB statement.

**CTJ401E** Rule Error: Type 1 invalid

*Explanation:* During scanning of the rule member, type 1 is detected, which is not one of the valid types.

*User Response:* Change the rule using the online interface.

**CTJ402E** Rule Error: Type 2 invalid when type 1 is R

*Explanation:* During scanning of the rule member, type 1 is detected, which is not one of the valid types.

*User Response:* Change the rule using the online interface.

**CTJ403E** Rule Error: ON\_RULE cards not in order

*Explanation:* During scanning of the rule member, an ON\_RULE statement with an invalid type was detected.

*User Response:* Change the rule using the online interface.
CTJ404E  Rule Error: ON criteria type is invalid

Explanation: During scanning of the rule member, an ON criteria statement with an invalid type was detected.
User Response: Change the rule using the online interface.

CTJ405E  Rule Error: ON criteria Type 3 invalid

Explanation: During scanning of the rule member, an ON criteria Type 3 statement with an invalid type was detected.
User Response: Change the rule using the online interface.

CTJ406E  Rule Error: ON criteria DDNAME card is missing

Explanation: During scanning of the rule member, missing card with DD not provided.
User Response: Change the rule using the online interface.

CTJ407E  Rule Error: DO_BLOCK type 2 is invalid

Explanation: During scanning of the rule member, type 2 of DO_BLOCK is not among valid types.
User Response: Change the rule using the online interface.

CTJ408E  Rule Error: DO_IF_BLOCK type 3 is invalid

Explanation: During scanning of the rule member, type 3 of DO IF is not among valid types.
User Response: Change the rule using the online interface.

CTJ409E  Rule Error: RETURN/REASON code not numeric/hexadecimal

Explanation: During scanning of the rule member a DO RETURN or EXIT return or reason code that is not numeric/hexadecimal respectively was detected.
User Response: Change the RETURN/REASON code using the online interface.

CTJ40AE  Rule Error: DO ELSE/END without previous DO IF

Explanation: During scanning of the rule member, a DO ELSE or DO END without a preceding DO IF was detected.
User Response: Change the DO-IF logic in the rule using the online interface.

CTJ40BI  Rule Error: string

Explanation: This message follows a rule error message and indicates the erroneous card.
User Response: No action is required.

CTJ40CI  Rule Loading. Member: member_name, Rule: rule_name

Explanation: Indicates loading of rule member.
User Response: No action is required.

CTJ40DE  Insufficient Storage

Explanation: Indicates that the rule buffer is not large enough.
User Response: Consult with your z/OS system administrator.

CTJ40EE  Abend S0abend_code at offset: offset in program code. Run string

Explanation: The CTJx11x program captured an abend S0Cxx.
User Response: Report this message to your system administrator.

CTJ40FE  RULES segments exceeds maximum

Explanation: The RULES segment exceeds maximum abend S0Cxx.
User Response: Consult with your z/OS system administrator.

CTJ40GI  string data pools has been successfully released

Explanation: At least one segment of RULES/Shared Memory data pool has been released.
User Response: No action is required.

CTJA01E  Control-M/JCL Verify is not installed

Explanation: The message is issued if Control-M/JCL Verify is not installed.
User Response: Verify that CTJ=Y is specified in the IOAPARM member and rerun Control-M/JCL Verify.

CTJA02W  Control-M/JCL Verify ended with a WARNING

Explanation: The message is issued when a warning message is issued by Control-M/JCL Verify for the job.
User Response: If required, modify the JCL of the job.

CTJA03E  Control-M/JCL Verify ended with an ERROR

Explanation: The message is issued when an error message is issued by Control-M/JCL Verify for the job.
User Response: Modify the JCL of the job.
CTJA04S    Control-M/JCL Verify ended with an ERROR. RC=return_code

Explanation: The message is issued when a severe error is encountered by Control-M/JCL Verify.

User Response: Refer to the Control-M/JCL Verify messages for a detailed explanation of the problem.

CTJD01W    jcl_statement_number WARNING: VOLUMES list and VOLUMES in catalog are not matched. DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the list of the volumes in the catalog is not match the list of volumes specified in the JCL. Wrong volumes list in the JCL that override the catalog list, might corrupt the data in the file.

User Response: Verify that the list of volumes specified in the JCL is correct.

CTJD02W    jcl_statement_number WARNING: DSN with DISP=NEW, already exists. DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the dataset, specified by DISP=NEW in the JCL, already exists.

User Response: Verify that the DISP and the DSN are correct.

CTJD03S    jcl_statement_number ERROR: CTJDSC GETMAIN error

Explanation: The message is issued when Control-M/JCL Verify starts validating the dataset, but fails to allocate workarea. The validation is aborted. This is an internal error.

User Response: Try again. If the problem reoccurs, call BMC Customer Support.

CTJD04S    jcl_statement_number ERROR: The DSN block was not found

Explanation: The message is issued when Control-M/JCL Verify starts validating the dataset, but fails to access the DSN block. The validation is aborted. This is an internal error.

User Response: Try again. If the problem reoccurs, call BMC Customer Support.

CTJD05W    jcl_statement_number WARNING: DSN with DISP=disp not found. DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the dataset specified by DISP=OLD or DISP=SHR does not exist, or has been deleted during the job processing.

User Response: Verify that the DISP and the DSN are correct.
CTJD06W  

jcl_statement_number WARNING: DSN with DISP=DELETE not found.  
DSN=dataset_name

**Explanation:** The message is issued when Control-M/JCL Verify detects that the dataset specified by DISP=DELETE does not exist, or has been previously deleted during the job processing.

**User Response:** Verify that the DISP and the DSN are correct.

CTJD07W  

jcl_statement_number WARNING: Syntax error, invalid DSN=dataset_name

**Explanation:** The message is issued when Control-M/JCL Verify detects that the dataset name is invalid because it does not comply with the syntax rules.

The validation continues without validating this dataset.

**User Response:** Provide a valid dataset name.

CTJD08W  

jcl_statement_number WARNING: file was already cataloged.  
DSN=dataset_name

**Explanation:** The message is issued when Control-M/JCL Verify starts the dataset validation and detects DISP CATLG. However, the file has already been cataloged so when the job is executed a "NOT CATLGD 2" message might be issued by the system.

**User Response:** Validate that the DSN and DISP=(xxx,CATLG) are correct.

CTJD09W  

jcl_statement_number WARNING: file was already cataloged on a previous statement. DSN=dataset_name

**Explanation:** The message is issued when Control-M/JCL Verify starts the dataset validation and detects DISP CATLG. However, the file has already been cataloged in a previous step so when the job is executed a "NOT CATLGD 2" message might be issued by the system.

**User Response:** Remove the CATLG from the DISP=(xxx,CATLG).

CTJD0AS  

jcl_statement_number The DSN TREE is empty

**Explanation:** The message is issued when Control-M/JCL Verify starts the dataset validation process, but detects that the list of datasets is empty.

The validation is aborted. This is an internal error.

**User Response:** Try again. If the problem reoccurs, call BMC Software Customer Support.
CTJD0BW  \textit{jcl\_statement\_number} WARNING: DSN'S DATA CLASS \textit{class} does not exist. \textit{DSN=dataset\_name}

\textit{Explanation}: The message is issued when Control-M/JCL Verify checks the data SMS class, but detects that the SMS class does not exist.

\textit{User Response}: Verify that the DATA CLASS is correct.

CTJD0CW  \textit{jcl\_statement\_number} WARNING: DSN'S STORAGE CLASS \textit{class} does not exist. \textit{DSN=dataset\_name}

\textit{Explanation}: The message is issued when Control-M/JCL Verify checks the storage SMS class, but detects that the SMS class does not exist.

\textit{User Response}: Verify that the STORAGE CLASS is correct.

CTJD0DW  \textit{jcl\_statement\_number} WARNING: DSN'S MANAGEMENT CLASS \textit{class} does not exist. \textit{DSN=dataset\_name}

\textit{Explanation}: The message is issued when Control-M/JCL Verify checks the management SMS class, but detects that the SMS class does not exist.

\textit{User Response}: Verify that the MANAGEMENT CLASS is correct.

CTJD0EW  \textit{jcl\_statement\_number} WARNING: VOLSER \textit{volser} is not in use by CATALOG \textit{DSN=dataset\_name}

\textit{Explanation}: The message is issued when Control-M/JCL Verify checks the VOLSER of a dataset, but detects that the VOLSER is different from the CATALOG information.

\textit{User Response}: Verify that the VOLSER and DSN are correct.

CTJD0FW  \textit{jcl\_statement\_number} WARNING: CATALOGED DSN, IS NOT FOUND. \textit{DSN=dataset\_name}

\textit{Explanation}: The message is issued when Control-M/JCL Verify checks if the dataset is cataloged, but it does not exist.

\textit{User Response}: Verify that the DSN is correct.

CTJD0GW  \textit{jcl\_statement\_number} WARNING: VOLSER \textit{volser} is not mounted

\textit{Explanation}: The message is issued when Control-M/JCL Verify checks the VOLSER of a dataset, but detects that the VOLSER is not mounted.

\textit{User Response}: Verify that the VOLSER is correct.
CTJD0HS  \text{\textit{jcl\_statement\_number}} WARNING: SMS CLASSES are used but SMS is not active. DSN=\textit{dataset\_name}

\textit{Explanation:} The message is issued when Control-M/JCL Verify detects the SMS CLASSES, while SMS is not active.

\textit{User Response:} Check if DFSMS is active and rerun the job after DFSMS has been started.

CTJD0IW  \text{\textit{jcl\_statement\_number}} WARNING: The specified SMS CLASS does not exist. DSN=\textit{dataset\_name}

\textit{Explanation:} The message is issued when Control-M/JCL Verify detects that either the SMS CLASS does not exist or it is invalid.

\textit{User Response:} Verify that the SMS CLASS is correct.

CTJD0JS  \text{\textit{jcl\_statement\_number}} DFSMS interface function ended not OK. DSN=\textit{dataset\_name}

\textit{Explanation:} The message is issued when Control-M/JCL Verify is retrieving information from the DFSMS interface and the process ends with an error.

\textit{User Response:} Check the previous messages from the SMS that might appear in the JOB LOG.

CTJD0KW  \text{\textit{jcl\_statement\_number}} WARNING: The specified UNIT=\textit{unit} does not exist.

\textit{Explanation:} The message is issued when Control-M/JCL Verify detects that the specified UNIT does not exist.

\textit{User Response:} Verify that the UNIT name is correct.

CTJD0LW  \text{\textit{jcl\_statement\_number}} WARNING: The specified UNIT=\textit{unit} is OFFLINE.

\textit{Explanation:} The message is issued when Control-M/JCL Verify detects that the specified UNIT is offline.

\textit{User Response:} Verify that the UNIT name is correct.

CTJD0MW  \text{\textit{jcl\_statement\_number}} WARNING: The specified VOLSER=\textit{volser} does not match UNIT=\textit{unit}

\textit{Explanation:} The message is issued when Control-M/JCL Verify detects that the specified VOLSER does not match the provided UNIT.

\textit{User Response:} Verify that the VOLSER and the UNIT are correct.
CTJD0NI  jcl_statement_number INFORMATION: DATASET is migrated. DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the dataset is migrated.

User Response: Verify that the dataset is correct.

CTJD0OW  jcl_statement_number WARNING: MEMBER= member_name is not found in DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the specified member does not exist in the specified dataset.

User Response: Verify that the member and the dataset are correct.

CTJD0PW  jcl_statement_number WARNING: Direct access to VSAM vsam of DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the dataset is a VSAM (index or data) component rather than a cluster.

User Response: Verify that the dataset is using the correct DSNAME.

CTJD0QW  jcl_statement_number WARNING: Using DISP=NEW but space was not provided. DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that DISP=NEW, but SPACE was not set in the JCL statement.

User Response: Verify if the SPACE setting is required or the space can be set by SMS ACS routines.

CTJD0RE  jcl_statement_number ERROR: DISP=NEW is not allowed in files' CONCATENATION. DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects DISP=NEW, but the DD statement is part of the CONCATENATION files. Since this is not valid, the job might abend due to an OPEN error.

User Response: Change the DISP=NEW to SHR or OLD, or remove the DD statement from the CONCATENATION group.

CTJD0TE  jcl_statement_number ERROR: DISP=OLD or SHR and VOL=SER was specified. UNIT is required for DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects the above DD statement and the UNIT keyword is missing. When DISP=OLD or SHR and VOL is specified, the dataset is accessed directly without checking the catalog and therefore the UNIT is required.

User Response: Correct the statement and rerun the job verification.
**CTJD0UE** 

`jcl_statement_number` ERROR: DISP=NEW and DISP=OLD for same file in the same step. DSN=*

*dataset_name*

**Explanation:** The message is issued when Control-M/JCL Verify detects the above DD statement and it detects that the dataset is referred to twice; once with DISP=NEW and once with DISP=SHR/OLD. The dataset will not be found because the dataset does not exist when the system tries to allocate the dataset as an existing dataset.

**User Response:** Correct the statement and rerun the job verification.

**CTJD0VE**

`jcl_statement_number` ERROR: DISP=(NEW,PASS) and twice DISP=(OLD,CATLG). DSN=*

*dataset_name*

**Explanation:** The message is issued when Control-M/JCL Verify detects the above DD statement and that the dataset is created with DISP=(NEW,PASS). In addition, it detects that the dataset is referred to twice in the same step with either DISP=(OLD,CATLG) or DISP=(SHR,CATLG). The second statement fails because the system removes the pass flag when handling the first DD statement. Refer to the MVS JCL Reference and MVS JCL Guide.

**User Response:** Correct the statement and rerun the job verification.

**CTJD0WE**

`jcl_statement_number` WARNING: USING DISP=NEW but space was not provided. DSN=*

*dataset_name*

**Explanation:** The message is issued when Control-M/JCL Verify detects that the DISP=NEW, but SPACE was not set in the JCL statement.

**User Response:** Add the missing SPACE keyword.

**CTJD0XW**

`jcl_statement_number` WARNING: DSN with DISP=NEW previously allocated in the job. DSN=*

*dataset_name*

**Explanation:** The message is issued when Control-M/JCL Verify detects that the dataset, specified by DISP=NEW in the JCL, was already allocated in a previous step.

**User Response:** Verify that the DISP and the DSN are correct.

**CTJD0YW**

`jcl_statement_number` WARNING: DSN with DISP=*

*disp_type*

was deleted in previous step. DSN=*

*dataset_name*

**Explanation:** The message is issued when Control-M/JCL Verify detects that the specified dataset has been deleted during the job processing.

**User Response:** Verify that the DISP and the DSN are correct.
CTJD0ZW  jcl_statement_number WARNING: DSN with SYMBOLICRELATE can not resolved. DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the dataset specified by ALIAS with SYMBOLICRELATE, but the symbol is not found in the system symbolic variables.

User Response: Verify that the SYMBOL can be resolved in the system.

CTJD10W  jcl_statement_number WARNING: DSN with DISP=disp_type was CREATED | DELETED in job jobname ID jobid statement statement_number [and may not exist | and may exist]

Explanation: The message is issued when Control-M/JCL Verify detects that the specified dataset was CREATED or DELETED in a previous job. When the current job and the previous job are not on the same level, the last section of the message, “and may not exist | and may exist,” is not relevant and therefore it is not displayed.

When the jobs are at the same level, the message changes according to the case as follows:

- The specified dataset was CREATED by a job at the same level, but since it might still be possible that the dataset does not yet exist, the warning message ends with the “and may not exist” phrase.
- The specified dataset was DELETED by a job at the same level, but since it might be possible that the dataset still exists, the warning message ends with the “and may exist” phrase.

User Response: Verify that the DISP and the dataset are correct.

CTJD11I  jcl_statement_number INFORMATION: DSN with DISP=PASS (attribute_number) is deleted. DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the file was created in the job with the PASS attribute, but was not cataloged or kept. The z/OS operation system deletes a file with the PASS attribute either during the last step of the job that references the file, or at the termination of the job.

Note: Control-M/JCL Verify does not issue this message for temporary datasets.

User Response: No action is required.

CTJD16W  jcl_statement_number WARNING: DSN is deleted will be used by job at the same level DISP=disposition job jobname ID job_id statement jcl_number

Explanation: The message is issued when Control-M/JCL Verify, using the job dependency awareness feature for Control-M jobs (VD=Y), detects that the dataset specified in jcl_number with DISP2=DELETE was used with DISP=SHR/OLD in a previous job job_id at the same job hierarchy level. The JOB might fail because the file might have been deleted.
User Response: Verify that the DISP and the DSN are correct.

CTJDG0I  jcl_statement_number INFORMATION: Allocating and concatenating all files in
GDG=gdg_base_name

Explanation: The message is issued when Control-M/JCL Verify detects that the
dataset, specified by DISP=SHR or OLD, is GDG and therefore all GDS in the GDG
will be allocated.

User Response: If this situation is not what is required, correct the DSNAME or add
the GDS relative number.

CTJDG1E  jcl_statement_number ERROR: Requested generation is above the limit of
gdg_limit. DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the
relative GDS number is higher than the GDG LIMIT in the GDG catalog entry. The
job will fail with a JCL error.

User Response: Correct the relative GDS number so that it is within the limit set in the
GDG catalog entry.

CTJDG2E  jcl_statement_number ERROR: Relative generation number specified for GDG
is larger than 255. DSN=dataset_name

Explanation: The message is issued when the Control-M/JCL Verify relative GDS
number is higher than 255. The job will fail with a JCL error.

User Response: 1) For DISP=SHR or OLD, correct the relative GDS number to be
within the limit set in the GDG catalog entry. 2) For DISP=NEW, set the relative GDS
number to a value lower than 255.

CTJDG3E  jcl_statement_number ERROR: Relative generation number is not numeric.
DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the
dataset is a GDS with a relative number, but the string number is not numeric. The
job will fail with a JCL error.

User Response: Correct the relative number.

CTJDG4E  jcl_statement_number ERROR: Relative generation number longer than three
characters. DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the
dataset is a GDS with a relative number, but the string contains more than three
characters. Since the maximum number of characters is three, the relative number is
invalid. The job will fail with a JCL error.

User Response: Correct the relative string number so that it is no more than three
characters.
WARNING: GDG is empty. No file will be allocated.
GDG=gdg_base_name

Explanation: The message is issued when Control-M/JCL Verify detects that the dataset is a GDG but does not contain any GDS. The job will fail with a JCL error.

User Response: Correct the name of the file or add a GDS to the GDG.

ERROR: GDS relative sign is not valid.
DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the GDS relative sign is not valid. (The valid values are: "+", "-" or "0".) The job will fail with a JCL error.

User Response: Correct the relative GDS number.

ERROR: Using existing GDS as a new file.
DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the dataset is a GDS with a relative number of zero or negative, but the DISP=NEW. The job will fail with a JCL error.

User Response: Correct the relative GDS number so that it is a positive number, or change the DISP=NEW to DISP=SHR/OLD/MOD.

ERROR: Using new GDS as an existing file.
DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the dataset is a GDS with a relative positive number (greater than zero), but the DISP=SHR or OLD. The job will fail with a JCL error.

User Response: Correct the relative GDS number so that it is zero or negative, or change the DISP=SHR/OLD/MOD to DISP=NEW.

ERROR: New GDS was previously allocated in this job.
DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the dataset is a GDS with DISP=NEW and with the same relative positive number that was already used in a previous step.

The job will fail with a JCL error.

User Response: Correct the GDS relative number to be a different positive number (greater than zero).
CTJDGAI  jcl_statement_number INFORMATION: DISP is disp. GDS will not ROLL INTO GDG. DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the dataset is a new GDS, but the termination DISP value is not CATLG and therefore, the GDS will be part of the GDG. In a SMS environment the file will be cataloged and the GDG will be marked as pending ROLE IN. A None-SMS file will not be cataloged and can exist on the volume.

User Response: Correct the DISP= to be DISP=(NEW,CATLG).

CTJDGBW  jcl_statement_number WARNING: DISP is UNCATLG. GDS will be removed from GDG. DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the dataset is a GDS, but the termination DISP=(xxx,UNCATLG) will cause the file to be uncataloged, removed from GDG, and deleted from the volume it is allocated on.

User Response: Correct the DISP= with the correct termination value.

CTJDGCE  jcl_statement_number ERROR: Using existing GDG as a new file. GDG=gdg_base_name

Explanation: The message is issued when Control-M/JCL Verify detects that the dataset is a GDG name with DISP=NEW in the JCL. A GDG cannot be a new file. A GDG with a relative positive number that is larger than zero can exist and it is called a GDS. The job will fail with a JCL error.

User Response: If the DSNAME is correct, add a GDS positive relative number that is greater than 0. Otherwise, change the DSNAME.

CTJDGDW  jcl_statement_number WARNING: Request GDS file but file is not a GDS type. DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the dataset is in a GDS format but it is not part of a GDG. The job will fail with a JCL error.

User Response: Provide a valid GDS or add the file to the GDG using the IDCAMS ALTER ROLLIN command.

CTJDGES  jcl_statement_number ERROR: GDS catalog search error. DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify starts the dataset validation and the internal validation failed. The validation is aborted.

User Response: Try again. If the problem reoccurs, call BMC Customer Support.
**CTJDGFE  jcl_statement_number** ERROR: GDS name is too long. max length is 35.
**DSN=dataset_name**

*Explanation:* The message is issued when Control-M/JCL Verify detects that the dataset is a GDS with a relative number, but the name is longer than 35 characters. When the GDG name, together with the relative number, is transformed to the full DSNAME, it cannot be longer than 44 characters. The relative number format ID ".GxxxxVyy". The job will fail with a JCL error.

*User Response:* Correct the dataset name.

**CTJDGGW  jcl_statement_number** WARNING: Request GDS file but GDG does not exist.
**GDG=gdg_base_name**

*Explanation:* The message is issued when Control-M/JCL Verify detects that the dataset is in a GDS format, but the GDG does not exist in the MVS Catalog. The job will fail with a JCL error.

*User Response:* Check whether the GDS name must be corrected.

**CTJE00E  alloc: dynalloc error message**

*Explanation:* Dynamic allocation or deallocation of the DSNAME specified in DO REXX failed with the given error message, where

- *alloc* is either ALLOC or DEALLOC.
- *dynalloc error message* is the IBM error message which starts with a message code of IKJ*.

This message is followed by message CTJE06I.

*System action:* For ALLOC: the DO REXX is not performed. For DEALLOC: the DO REXX has been performed.

*User Response:* For ALLOC, correct the DSNAME in the DO REXX. For DEALLOC, contact BMC support.

**CTJE01E  DO REXX failed with IRXEXEC RC=return_code**

*Explanation:* The call to IRXEXEC to execute REXX EXEC failed with RC=return_code.

*User Response:* To understand the reason for the failure, note the following:

- Return codes of IRXEXEC are described in the z/OS V1R13.0 TSO/E REXX Reference.
- RC=20 usually means that the specified EXEC (member) was not found in the specified DSN.
- Review additional IRX* error messages in JESMSGLG.
- RC > 20000 means that there was a syntax error in the EXEC.

- See SYSTSPRT for additional IRX* error messages.

This message is followed by message CTJE06I.
If the reason for the failure is still not clear, contact BMC support.

**CTJE02E**  DOMSG from EXEC: No message provided

*Explanation:* DOMSG did not provide the message text.
*User Response:* Correct the error described above.

**CTJE03E**  DORC from EXEC: No RC and no RS provided

*Explanation:* DORC did not provide the return code (RC) and the reason code (RS). This message is followed by message CTJE06I.
*User Response:* Correct the error described above.

**CTJE04E**  DORC from EXEC: Invalid decimal characters in RC=return_code

*Explanation:* The return code (RC) in DORC must contain only decimal characters.
*User Response:* Correct the error described above.

**CTJE05E**  DORC from EXEC: Invalid hex characters in RS=reason_code

*Explanation:* The reason code (RS) in DORC must contain only hex characters. This message is followed by message CTJE06I.
*User Response:* Correct the error described above.

**CTJE06I**  RULE=rule name TABLE=table name EXEC=exec DSN=dataset_name

*Explanation:* This message identifies which rule and EXEC caused the error described in the preceding message.
*User Response:* Correct the error described above.

**CTJE07E**  CTJRXX SETVAR or RESOLVE was called with incorrect number of arguments.

*Explanation:* A REXX EXEC invoked by a DO REXX includes a call to CTJRXX SETVAR or RESOLVE with more or less than 2 arguments.

- CTJRXX SETVAR and RESOLVE must be called with 2 arguments. This message is followed by message CTJE06I.

- The SETVAR or RESOLVE function is not performed.
*User Response:* Correct the CTJRXX call in the REXX EXEC.
CTJJ00I SITE STANDARD'S RULES WERE NOT LOADED, SITE STANDARDS VER. DEACTIVATED

*Explanation:* The site standards rules were not loaded, therefore, this kind of verification is deactivated for this run.

*User Response:* Load the rules using the CTJINIT utility.

CTJJ01E ERROR: Missing statement *begin_statement* before *block_type*

*Explanation:* While verifying a JES ENDPROUTE or ENDDATASET statement, Control-M/JCL Verify detected that a PROCESS or a DATASET statement is missing before the beginning of the PROCESS or DATASET block.

*User Response:* Either add the missing statement or remove the above statement and then rerun the job verification.

CTJJ02E ERROR: SIGNON ERROR - NEW PASSWORD is incorrect, must be in position 35 to 42

*Explanation:* While verifying the SIGNON JES statement, Control-M/JCL Verify detected that the new password is incorrect.

*User Response:* Correct the password and rerun the job verification.

CTJJ03E ERROR: SIGNON ERROR - NEW PASSWORD2 is incorrect, must be in position 73 to 80

*Explanation:* While verifying the SIGNON JES statement, Control-M/JCL Verify detected that the new password2 is incorrect.

*User Response:* Correct the password and rerun the job verification.

CTJJ04E ERROR: Invalid JES statement - placement in JCL is not correct

*Explanation:* This message is issued when Control-M/JCL Verify detects that a JES2 XMIT or JES3 ROUTE statement is incorrectly placed in the JCL.

*User Response:* Ensure that the JES2 XMIT or JES3 ROUTE statement is placed correctly after the JOB statement.

CTJJ05E ERROR: SIGNON ERROR - blanks are required after WORKSTATION name until position 21

*Explanation:* While verifying the SIGNON JES statement, Control-M/JCL Verify detected that blanks are missing after the workstation name.

*User Response:* Add the required blanks after the workstation name, until position 21, and rerun the job verification.
CTJJ06E   ERROR: SIGNON ERROR - BLANK is required in position 24

Explanation: While verifying the SIGNON JES3 statement, Control-M/JCL Verify detected that position 24 is not blank.
User Response: Correct the statement and rerun the job verification.

CTJJ07E   ERROR: SIGNON ERROR - ‘A’ or BLANK is required in position 22

Explanation: While verifying the SIGNON JES statement, Control-M/JCL Verify detected that neither a “A” nor a blank is in position 22.
User Response: Correct the statement and rerun the job verification.

CTJJ08E   ERROR: SIGNON ERROR - ‘R’ or BLANK is required in position 23

Explanation: While verifying the SIGNON JES statement, Control-M/JCL Verify detected that neither a “R” nor a blank is in position 23.
User Response: Correct the statement and rerun the job verification.

CTJJ09E   INTERNAL ERROR CODE=100

Explanation: While verifying a JES statement, Control-M/JCL Verify detected an internal error.
User Response: Rerun the job verification. If the problem reoccurs, call BMC Customer Support.

CTJJ09W   WARNING: The above /*SETUP statement has no volume to request

Explanation: While verifying the SETUP JES2 statement, Control-M/JCL Verify detected that the statement does not contain any VOLUME for making a mounting request to the OPERATOR. JES2 does not issue a $HASP103 null, but the JES2 sets the JOB to hold state.
User Response: Correct the statement and rerun the job verification.

CTJJ0AE   ERROR: Above JES statement has a keyword error

Explanation: While verifying a JES statement, Control-M/JCL Verify detected invalid data in the keyword statement.
User Response: Correct the keyword data and rerun the job verification.

CTJJ0BE   ERROR: JES statement node is unknown to JES. NODE:node_string

Explanation: While verifying a JES statement, Control-M/JCL Verify detected a node that is not known to JES.
User Response: Correct the node and rerun the job verification.
CTJJ0CI   INFORMATION: Keyword was ignored by Control-M/JCL Verify

Explanation: While verifying a JES statement, Control-M/JCL Verify ignored the keyword.
User Response: No action is required.

CTJJ0DE   ERROR: Missing mandatory keyword

Explanation: While verifying a JES statement, Control-M/JCL Verify detected that a mandatory keyword is missing for the above type of JES statement. For more information, see the IBM MVS JCL Reference.
User Response: Add the missing keyword and rerun the job verification.

CTJJ0EW   WARNING: KEYWORD keyword has invalid value: string

Explanation: While verifying a JES statement, Control-M/JCL Verify detected that the keyword statement contains an invalid value.
User Response: Correct the keyword data and rerun the job verification.

CTJJ0FE   ERROR: Invalid JES statement type

Explanation: While verifying a JES statement, Control-M/JCL Verify detected an invalid statement type.
User Response: Correct the statement type and rerun the job verification.

CTJJ0GE   ERROR: Invalid keyword: keyword

Explanation: While verifying a JES statement, Control-M/JCL Verify detected an invalid keyword.
User Response: Correct the keyword and rerun the job verification.

CTJJ0HE   ERROR: Invalid delimiter: delimiter

Explanation: While verifying a JES statement, Control-M/JCL Verify detected an invalid delimiter between two keywords or values.
User Response: Correct the delimiter and rerun the job verification.

CTJJ0IE   ERROR: Multiply-defined keyword: keyword

Explanation: While verifying a JES statement, Control-M/JCL Verify detected a keyword used more than one time in the statement.
User Response: Correct the statement and rerun the job verification.
**CTJJ0JE**  ERROR: Keyword *keyword* data is not numeric. DATA:*data*

*Explanation:* While verifying a JES statement, Control-M/JCL Verify detected that the keyword data is not numeric, as required by JES.

*User Response:* Correct the keyword and rerun the job verification.

**CTJJ0KE**  ERROR: Keyword *keyword* VALUE *value* is out of range

*Explanation:* While verifying a JES statement, Control-M/JCL Verify detected that the keyword data is not in the allowed range.

*User Response:* Correct the keyword data and rerun the job verification.

**CTJJ0LE**  ERROR: Total number of groups is higher than 255

*Explanation:* While verifying a JES statement, Control-M/JCL Verify detected that the total number of groups is greater than 255.

*User Response:* Correct the groups and rerun the job verification.

**CTJJ0ME**  ERROR: Too many groups in list. Maximum allowed is 8 groups.

*Explanation:* While verifying a JES statement, Control-M/JCL Verify detected that the number of groups is greater than the maximum allowed by JES.

*User Response:* Correct the groups and rerun the job verification.

**CTJJ0NI**  INFORMATION: Invalid PRIORITY statement - ignored by JES2

*Explanation:* While verifying a JES statement, Control-M/JCL Verify detected that the PRIORITY statement is invalid. JES2 will ignore the PRIORITY statement when the job is submitted.

*User Response:* If you want to use the PRIORITY statement, correct the PRIORITY statement and rerun the job verification.

**CTJJ0OE**  ERROR: SIGNON Invalid REMOTE WORKSTATION:*string*

*Explanation:* While verifying a JES statement, Control-M/JCL Verify detected that the remote workstation name in the SIGNON statement is invalid.

*User Response:* Correct the statement and rerun the job verification.

**CTJJ0PI**  INFORMATION: Multiple //*NET statements encountered in input stream

*Explanation:* While verifying a JES statement, Control-M/JCL Verify detected multiple //*NET statements.

*User Response:* Correct the JCL and rerun the job verification.
CTJJ0QI INFORMATION: JES statement above may have an invalid TSO or z/VM USERID:userid

Explanation: While verifying a JES statement, Control-M/JCL Verify detected an invalid TSO or VM user ID.

User Response: If the remote destination is a z/OS or a z/VM you might have to correct the user ID.

CTJJ0RE ERROR: Unbalanced parentheses on keyword data

Explanation: While verifying a JES statement, Control-M/JCL Verify detected unbalanced parentheses on keyword data.

User Response: Correct the parentheses on the keyword data and rerun the job verification.

CTJJ0SE ERROR: SIGNON ERROR - Must have BLANKS in position 9 to 15

Explanation: While verifying a SIGNON statement, Control-M/JCL Verify detected that the data between position 9 and 15 is not BLANK, as required.

User Response: Correct the SIGNON statement and rerun the job verification.

CTJJ0TE INTERNAL ERROR

Explanation: While verifying a JES statement, Control-M/JCL Verify detected an internal error.

User Response: Rerun the job verification. If the problem reoccurs, call BMC Customer Support.

CTJJ0UE ERROR: SIGNON ERROR - Must be a REMOTE name group position 16 to 24

Explanation: While verifying a SIGNON statement, Control-M/JCL Verify detected that the data of the REMOTE name field is blank.

User Response: Correct the SIGNON statement and rerun the job verification.

CTJJ0VE ERROR: Keyword keyword invalid destination name.

Explanation: While verifying a JES statement, Control-M/JCL Verify detected that the destination name is not valid.

User Response: Correct the keyword and rerun the job verification.

CTJJ0WE ERROR: SIGNON ERROR - PASSWORD1 invalid, must be in position 25 to 32

Explanation: While verifying a SIGNON statement, Control-M/JCL Verify detected that the password1 is invalid.

User Response: Correct the password and rerun the job verification.
CTJJ0XE  ERROR: *keyword*/statement_string has invalid value:*value*

Explanation: While verifying a JES statement, Control-M/JCL Verify detected that the keyword has an invalid value.

User Response: Correct the keyword and rerun the job verification.

CTJJ0YE  ERROR: Missing or invalid DDNAME

Explanation: While verifying a JES statement, Control-M/JCL Verify detected that the mandatory keyword DDNAME is missing or has an invalid value.

User Response: Correct the keyword and rerun the job verification.

CTJJ0ZE  ERROR: Keyword *keyword*/statement_string data length error

Explanation: While verifying a JES statement, Control-M/JCL Verify detected that the keyword’s data length is not valid.

User Response: Correct the keyword and rerun the job verification.

CTJJ11E  ERROR: SIGNON ERROR - NEW PASSWORD2 is incorrect, must be in position 35 to 42

Explanation: While verifying the SIGNON JES statement, Control-M/JCL Verify detected that the new password2 is incorrect.

User Response: Correct the password and rerun the job verification.

CTJJ12E  ERROR: SIGNON ERROR - NEW PASSWORD is incorrect, must be in position 44 to 51

Explanation: While verifying the SIGNON JES statement, Control-M/JCL Verify detected that the new password2 is incorrect.

User Response: Correct the password and rerun the job verification.

CTJJ13W  WARNING: KEYWORD *keyword* member for value *string* does not exist in SYS1.IMAGELIB

Explanation: While verifying a JES statement, Control-M/JCL Verify detected that the keyword statement contains an invalid value.

User Response: Correct the keyword data and rerun the job verification.

CTJJ14I  INFORMATION: PRIORITY value *priority_string* is not numeric. Statement will be ignored by JES2

Explanation: While verifying a JES statement, Control-M/JCL Verify detected that the PRIORITY value is not numeric, as required. JES2 will ignore the PRIORITY statement when the job is submitted.
User Response: If you want to use the PRIORITY statement, correct the PRIORITY statement and rerun the job verification.

CTJJ15I  INFORMATION: PRIORITY value priority_number is out of range. Statement will be ignored by JES2

Explanation: While verifying a JES statement, Control-M/JCL Verify detected that the PRIORITY value is not between 0 to 15. JES2 will ignore the PRIORITY statement when the job is submitted.

User Response: If you want to use the PRIORITY statement, correct the PRIORITY statement and rerun the job verification.

CTJJ18E  ERROR: unknown NODE/REMOTE/DEST name

Explanation: While verifying a JES statement, Control-M/JCL Verify detected that a NODE/REMOTE/DEST name is unknown to JES.

User Response: Correct the statements and rerun the job verification.

CTJP0CW jcl_statement_number WARNING: PGM pgm_module not found

Explanation: The message is issued when Control-M/JCL Verify detects that the specified PGM module does not exist.

User Response: Check if the module name is correct. If necessary, add a STEPLIB or JOBLIB.

CTJP0EI jcl_statement_number INFORMATION: PGM pgm_module found in the STEPLIB library

Explanation: The message is issued when Control-M/JCL Verify detects that the specified PGM module exists in the given STEPLIB library.

User Response: No action is required.

CTJP0FI jcl_statement_number INFORMATION: PGM pgm_module found in the JOBLIB library

Explanation: The message is issued when Control-M/JCL Verify detects that the specified PGM module exists in the given JOBLIB library.

User Response: No action is required.

CTJP0GI jcl_statement_number INFORMATION: PGM pgm_module found in LINKLIST library

Explanation: The message is issued when Control-M/JCL Verify detects that the specified PGM module exists in one of the given LINKLIST libraries.

User Response: No action is required.
CTJP0HI  \textit{jcl\_statement\_number IN\textsc{formation}}: PGM \textit{pgm\_module} found in LPA/PLPA library

\textit{Explanation}: The message is issued when Control-M/JCL Verify detects that the specified PGM module exists in one of the given LPA or PLPA libraries.

\textit{User Response}: No action is required.

CTJP0II \textit{jcl\_statement\_number IN\textsc{formation}}: PGM \textit{pgm\_module} found in library

\textit{Explanation}: The message is issued when Control-M/JCL Verify detects that the specified PGM module exists in the given library.

\textit{User Response}: No action is required.

CTJR01I Control-M/JCL Verify started resolving Control-M AutoEdit variables

\textit{Explanation}: The message is issued when Control-M/JCL Verify begins resolving Control-M AutoEdit variables.

\textit{User Response}: No action is required.

CTJR02I \textit{AutoEdit variable list}

\textit{Explanation}: The message is issued when Control-M/JCL Verify is resolving Control-M AutoEdit variables. The text contains a list of the Control-M AutoEdit variables being resolved.

\textit{User Response}: No action is required.

CTJR03I Control-M/JCL Verify completed resolving Control-M AutoEdit variables

\textit{Explanation}: The message is issued when Control-M/JCL Verify finishes resolving Control-M AutoEdit variables.

\textit{User Response}: No action is required.

CTJR04I Control-M/JCL Verify resolving JOB=\textit{jobname} in LIB=\textit{library}

\textit{Explanation}: The message is issued when Control-M/JCL Verify is resolving the Control-M AutoEdit variables in the \textit{jobname} job located in the \textit{library} library.

\textit{User Response}: No action is required.

CTJR05I \textit{text}

\textit{Explanation}: The message is issued when Control-M/JCL Verify is resolving Control-M AutoEdit variables and replaces the original JCL statement with a resolved JCL statement. The text contains the original JCL statement.

\textit{User Response}: No action is required.
CTJR06E  Job definition(s) not supported by JCL verification

*Explanation:* The message is issued when Control-M/JCL Verify finds a schedule definition without one of the following Task Types: Regular job, Emergency job, or Cyclic job.

*User Response:* No action is required.

CTJR07E  Empty JOB *jobname* in MEMBER *member_name* in LIBRARY *lib_name*

*Explanation:* The message is issued when Control-M/JCL Verify attempts to verify a JCL member, which is specified in the *lib_name* library, but the *member_name* member is empty.

*User Response:* Check the member and library names, and correct the incorrect name.

CTJR08I  Verify JCL in JOB *jobname* from TABLE *table* in LIBRARY *library*

*Explanation:* The message is issued when Control-M/JCL Verify is verifying the *jobname* job from the *table* table, located in the *library* library.

*User Response:* No action is required.

CTJR09I  Verify JCL for JOB *jobname* will not be verified since it is *reason*.

*Explanation:* The *jobname* job will not be verified by BMC Control-M Workload Automation JCL Verify because of the reason indicated by *reason*.

*User Response:* No action is required.

CTJR0AI  NO JOB *jobname* WAS FOUND IN TABLE *table*

*Explanation:* The message is issued when Control-M/JCL Verify attempts to verify a job, but fails to locate the *jobname* job in the *table* table. This an internal error.

*User Response:* No action is required.

CTJS00I  INFORMATION: JCL contains no errors

*Explanation:* The message is issued when Control-M/JCL Verify does not detect any errors in the JCL.

*User Response:* No action is required.

CTJS01E  ERROR: JCL errors found

*Explanation:* The message is issued when Control-M/JCL Verify detects errors in the JCL.

*User Response:* Read the error messages to determine the problem.
CTJS02E  Unable to open PROCLIB DDNAME

*Explanation:* The message is issued when Control-M/JCL Verify is unable to open the JES2 or JES3 PROCLIB library, indicated by the `ddname` DD statement, to access the JCL procedure.

*User Response:* Check previous messages and additional messages in the JOBLOG.

CTJS03E  ERROR: Invalid or missing JOB statement

*Explanation:* The message is issued when Control-M/JCL Verify detects that the JCL contains an invalid or missing JOB statement.

*User Response:* Correct the JOB statement or add one in the JCL member.

CTJS04E  Invalid parameter list

*Explanation:* The message is issued when Control-M/JCL Verify receives an invalid parameter list.

*User Response:* Contact BMC Software Customer Support.

CTJS05E  ERROR: Converter/Interpreter failure

*Explanation:* The message is issued when Control-M/JCL Verify had an internal failure.

*User Response:* Contact BMC Software Customer Support.

CTJS06E  `jcl_statement_number` ERROR: Invalid REFERBACK in the DSN=`dataset_name`

*Explanation:* The message is issued when Control-M/JCL Verify detects an error with the BACK REFERENCE in the DSN keyword.

*User Response:* Check the BACK REFERENCE DD statement to determine the problem.

CTJS07E  `jcl_statement_number` ERROR: Invalid REFERBACK in the VOL=REF=`string`

*Explanation:* The message is issued when Control-M/JCL Verify detects an error with the BACK REFERENCE in the VOL=REF keyword.

*User Response:* Check the BACK REFERENCE DD statement to determine the problem.

CTJS08E  `jcl_statement_number` ERROR: Invalid REFERBACK in the UNIT=AFF=`string`

*Explanation:* The message is issued when Control-M/JCL Verify detects an error with the BACK REFERENCE in the UNIT=AFF keyword.

*User Response:* Check the BACK REFERENCE DD statement to determine the problem.
CTJS09E  jcl_statement_number ERROR: Invalid REFERBACK in the PGM=string

Explanation: The message is issued when Control-M/JCL Verify detects an error with the BACK REFERENCE in the PGM= keyword.

User Response: Check the BACK REFERENCE DD statement to determine the problem.

CTJS0AI  jcl_statement_number INFORMATION: DDNAME ddname previously allocated in this step

Explanation: The message is issued when Control-M/JCL Verify detects that the specified DDNAME was previously allocated in the STEP.

User Response: No action is required.

CTJS0BE  jcl_statement_number ERROR: Invalid REFERBACK in the DCB=string

Explanation: The message is issued when Control-M/JCL Verify detects an error with the BACK REFERENCE in the DCB= keyword.

User Response: Check the BACK REFERENCE DD statement to determine the problem.

CTJS0CW  jcl_statement_number WARNING: Inconsistent dataset type and directory block specification

Explanation: The message is issued when Control-M/JCL Verify detects that the user specified DSORG=PO without the directory block parameter.

User Response: No action is required.

CTJS0DW  jcl_statement_number WARNING: Record length of zero is only valid if RECFM=U

Explanation: The message is issued when Control-M/JCL Verify detects that the user specified LRECL=0 with RECFM=U.

User Response: No action is required.

CTJS0EW  jcl_statement_number WARNING: Block size must be a multiple of record length for RECFM=FB

Explanation: The message is issued when Control-M/JCL Verify detects that the user specified LRECL=0 with RECFM set to a value other than U.

User Response: Change the LRECL setting, or set RECFM=U, and rerun the job verification.
**WARNING: Record length must equal blocksize for RECFM=F**

*Explanation:* The message is issued when Control-M/JCL Verify detects that the user specified LRECL=BLKSIZE with RECFM=F.

*User Response:* No action is required.

**WARNING: Record length must be less than block size for RECFM=F**

*Explanation:* The message is issued when Control-M/JCL Verify detects that the user specified LRECL with a value greater than BLKSIZE for RECFM=F.

*User Response:* No action is required.

**WARNING: Block size inconsistent with record size for RECFM=V**

*Explanation:* The message is issued when Control-M/JCL Verify detects that the user specified LRECL with a value greater than (BLKSIZE-4) for RECFM=F.

*User Response:* No action is required.

**INFORMATION: Block size zero or missing, will be set by the system**

*Explanation:* The message is issued when Control-M/JCL Verify detects that the block size setting is missing or the block size is set to zero.

*User Response:* No action is required.

**INFORMATION: Unable to open PROCLIB DD statement**

*Explanation:* The message is issued when Control-M/JCL Verify is unable to open the JES2 or JES3 PROCLIB library, indicated by the *ddname* DD statement, to access the JCL procedure.

*User Response:* Check previous messages and additional messages in the JOBLOG.

**IOA security is disabled. Access validation is disabled.**

*Explanation:* The message is issued when Control-M/JCL Verify detects during initialization that the IOA security is either disabled or not installed. Therefore, the Control-M/JCL Verify file access validation feature is disabled.

*User Response:* Contact your IOA administrator to ensure that IOA security is installed and enabled.
**CTJSA1E**  ERROR: Permission to access was not granted for DSN=dataset_name

*Explanation:* The message is issued when Control-M/JCL Verify detects that the user that performs the verification does not have READ access to the JCL library that contains the jobs being verified. The verification request is aborted.

*User Response:* Verify that the provided user has READ access to the JCL library.

**CTJSA1W**  jcl_statement_number WARNING: Access was not granted for DSN=dataset_name

*Explanation:* The message is issued when Control-M/JCL Verify detects that access was not granted to the provided user for the indicated dataset.

*User Response:* Verify that the provided user has sufficient authority for the dataset.

**CTJSA2W**  jcl_statement_number WARNING: Permission to READ was not granted for DSN=dataset_name

*Explanation:* The message is issued when Control-M/JCL Verify detects that read access was not granted to the provided user for the indicated dataset.

*User Response:* Verify that the correct user is being used and that the user is allowed to read the dataset.

**CTJSA3W**  jcl_statement_number WARNING: Permission to UPDATE was not granted for DSN=dataset_name

*Explanation:* The message is issued when Control-M/JCL Verify detects that update access was not granted to the provided user for the indicated dataset.

*User Response:* Verify that the correct user is being used and that the user is allowed to update the dataset.

**CTJSA4S**  jcl_statement_number WARNING: No authorization to commit a security query. DSN=dataset_name

*Explanation:* The message is issued when Control-M/JCL Verify detects that authority for committing a security query was not granted to the provided user for the indicated dataset.

*User Response:* Check if the STEPLIB or JOBLIB library are APF authorized. Retry, and if the problem is not solved, call BMC Software Customer Support.

**CTJSA5W**  jcl_statement_number WARNING: Permission to delete was not granted for DSN=dataset_name

*Explanation:* The message is issued when Control-M/JCL Verify detects that ALTER access was not granted to the provided user for the indicated dataset and therefore the user cannot delete the dataset.
User Response: Verify that the correct user is being used and that the user has ALTER authority for the dataset.

CTJSA6W  
`jcl_statement_number WARNING: Permission to UNCATLG was not granted for DSN=dataset_name`

Explanation: The message is issued when Control-M/JCL Verify detects that ALTER access was not granted to the provided user for the indicated dataset and therefore the user cannot uncatalog the dataset.

User Response: Verify that the correct user is being used and that the user is allowed to uncatalog the dataset.

CTJSA7W  
`WARNING: The provided user is not defined. USERID=user_ID`

Explanation: The message is issued when Control-M/JCL Verify detects that the provided user is not defined in the SECURITY package (CA-ACF2, CA/Top Secret or IBM RACF).

User Response: Verify that the correct user is being used.

CTJSA8W  
`Access validation check cannot be performed by Control-M/JCL Verify`

Explanation: The message is issued when Control-M/JCL Verify detects that the provided user does not have a valid user ID and therefore the access validation check cannot be performed.

User Response: Check for previous messages in the SYSPRINT or JOBLOG.

CTJSA9I  
`Access Verification USERID is changed from string_1 to string_2`

Explanation: Control-M/JCL Verify displays this message when the access verification user ID has been changed to the user ID that appears in the scanned job statement.

User Response: No action is required.

CTJT001I  
`Only ENTER key is valid`

Explanation: When Control-M/JCL Verify Edit Macro is executed for a member that has been changed, but not saved, a prompt window opens, asking if the user wants to save the changes. If the user presses any key, besides ENTER, first ISPF displays the following short message:

Invalid PF key

If the user presses PF1, ISPF displays the full CTJT001I message.

User Response: In the prompt window, enter "Y" or "N" and press ENTER.
CTJT002I  Valid values are I, W, E or D

Explanation: When Control-M/JCL Verify Edit Macro is executed with the “P” parameter, a prompt window opens, allowing the user to change parameter values. If the user enters an invalid value for the MESSAGE LEVEL field, first ISPF displays the following short message:

Invalid msg level value

If the user presses PF1, ISPF displays the full CTJT002I message.

User Response: Set the message level, by entering one of the valid values (I, W, E or D) in the MESSAGE LEVEL field.

CTJT003I  Valid values are Y, N, or D

Explanation: When Control-M/JCL Verify Edit Macro is executed with the “P” parameter, a prompt window opens, allowing the user to change parameter values. If the user enters an invalid value for one or more of the following fields: JES SYNTAX, DATASET AUTHORIZED, DATASET EXISTENCE or PROGRAM EXISTENCE, first ISPF displays the following short message:

Invalid parm value

If the user presses PF1, ISPF displays the full CTJT003I message.

User Response: Set the parameter value, by entering one of the valid values (Y, N, or D) in the appropriate field.

CTJT004I  USERID field can not be empty

Explanation: When Control-M/JCL Verify Edit Macro is executed with the “P” parameter, a prompt window opens, allowing the user to change parameter values. If the user enters an empty value for the USERID field, first ISPF displays the following short message:

USERID field empty

If the user presses PF1, ISPF displays the full CTJT004I message.

User Response: Enter a valid user ID in the USERID field.

CTJT005I  USERID should be a valid user name

Explanation: When Control-M/JCL Verify Edit Macro is executed with the “P” parameter, a prompt window opens, allowing the user to change parameter values. If the user enters an invalid value for the USERID field, first ISPF displays the following short message:

Invalid USERID field

If the user presses PF1, ISPF displays the full CTJT005I message.

User Response: Enter a valid user ID in the USERID field.
CTJU00I  Control-M/JCL Verify Utility started. LEVEL apar_id

Explaination: The message is issued when Control-M/JCL Verify starts the first time. Level indicates the module APAR ID.

User Response: No action is required.

CTJU01I  echo-sysin

Explaination: The message echoes the input from PARM or *SYSIN statements to Control-M/JCL Verify.

User Response: No action is required.

CTJU02I  Control-M/JCL Verify Utility ended. RC=return_code.

Explaination: The message is issued when Control-M/JCL Verify ends. RC is the highest return code that occurs during the validation processing.

User Response: If RC is not zero, it means that at least one validation has failed. Review the warning and error messages and handle them appropriately.

CTJU03I  Input not read from SYSIN. RC=12

Explaination: The message is issued when Control-M/JCL Verify attempts to read the SYSIN file, but it is either an empty or a DUMMY file.

User Response: Add at least one valid statement or a comment and rerun the job verification.

CTJU0CE Keyword keyword is not a valid keyword

Explaination: The message is issued when Control-M/JCL Verify detects an unrecognized keyword, and continues to the next input statement. Control-M/JCL Verify terminates with RC=12.

User Response: Correct the keyword and re-submit the Control-M/JCL Verify job.

CTJU0EE Input library not found.

Explaination: The message is issued after Control-M/JCL Verify scans all statements, but detects that a SCAN or ORDER statement does not have a DSN or LIBRARY keyword or the value is null. Control-M/JCL Verify terminates with RC=12.

User Response: Add a DSN or LIBRARY keyword with a file name to the SCAN or ORDER statement and re-submit the Control-M/JCL Verify job.

CTJU0FE Function function Is not a valid function

Explaination: The message is issued after Control-M/JCL Verify detects a statement containing an invalid function. Control-M/JCL Verify terminates with RC=12.

User Response: Correct the statement and re-submit the Control-M/JCL Verify job.
CTJU0GE  Keyword *keyword* contains invalid data *data*

*Explanation:* The message is issued after Control-M/JCL Verify detects that a keyword in one of the statements contains invalid data. Control-M/JCL Verify terminates with RC=12.

*User Response:* Correct the keyword data and re-submit the Control-M/JCL Verify job.

CTJU0HE  The requested ORDERID *text*

*Explanation:* A job, which is requested by either the ORDERID or the AJF function, is not processed because of one of the following reasons: it does not have embedded JCL statements, it is a TABLE Entity, it is a Started Task, it is not found in the AJF, or it is already deleted.

*User Response:* Verify that the requested job is correct.

CTJUV0I  INFORMATION: verifying utility *util_name*

*Explanation:* The message is issued when Control-M/JCL Verify starts verifying a utility program.

*User Response:* No action is required.

CTJUV1I  INFORMATION: verifying utility *util_name* ended RC=*return_code*.

*Explanation:* The message is issued when Control-M/JCL Verify ended the utility program verification with a return code.

*User Response:* If the return code is not 0, correct the error as specified in the previous messages.

CTJUV2I  jcl_statement_number INFORMATION: *dd_name* is DUMMY. *util_name* messages will not be available.

*Explanation:* The message is issued when Control-M/JCL Verify detects the utility DD messages' DD statement is DUMMY or NULLFILE and during the utility runtime the messages will not be listed.

*User Response:* If this is correct, no change is needed. Otherwise, set the DD name to a SYSOUT or to a dataset.

CTJUV3I  jcl_statement_number INFORMATION: *dd_name* is DUMMY. *util_name* will use defaults.

*Explanation:* The message is issued when Control-M/JCL Verify detects the utility control statement input data set is DUMMY or NULLFILE. Therefore the utility will use the default input statements and parameters.

*User Response:* If this is correct, no change is needed. Otherwise, set the DD name to a SYSIN or to a dataset with the required control statements.
CTJUV4S  **util_name** ERROR: LOAD error *pgm_name*. Verifying is terminated.

*Explanation:* The message is issued when Control-M/JCL Verify fails to load the utility program. The verification is terminated.

Control-M/JCL Verify uses standard LOAD or LINK services to invoke the utility.

*User Response:* Verify that the *pgm_name* exists in the verification JOBLIB, STEPLIB or one of the system LINKLIST libraries.

CTJUV5E  **util_name** ERROR: Label error.

*Explanation:* The message is issued when Control-M/JCL Verify detects that the previous statement contains an invalid label.

*User Response:* Correct the label.

CTJUV6I  **jcl_statement_number** INFORMATION: *dd_name* - *util_name* is empty.

**MEMBER=** *member_name*  **DSN=** *dataset_name*

*Explanation:* The message is issued when Control-M/JCL Verify read that the *util_name* and the *dataset_name* or *member_name* is empty.

*User Response:* No action is required.

CTJUV7I  **jcl_statement_number** INFORMATION: *dd_name* concatenation contains DUMMY DD. All the following concatenated data sets will be ignored.

*Explanation:* The message is issued when Control-M/JCL Verify detects the utility control statement input datasets’ concatenation contains a DUMMY statement or DSN=NULLFILE statement. Therefore the utility will ignore all the datasets following this statement.

*User Response:* If this is correct, no change is needed. Otherwise, set the DD name to a SYSIN or to a dataset with the required control statements.

CTJUV8E  **util_name** ERROR: Label error.

*Explanation:* The message is issued when Control-M/JCL Verify detects that the previous statement contains a label that is too long.

*User Response:* Correct the label.

CTJUV9E  **util_name** ERROR: Null statement.

*Explanation:* The message is issued when Control-M/JCL Verify detects that the previous statement does not contain any text besides the label.

*User Response:* Delete the statement.
CTJUVAE  
*util_name* ERROR: Missing mandatory DD statement *dd_name*

*Explanation:* The message is issued when Control-M/JCL Verify detects that a mandatory DD statement is missing in the STEP that calls the utility program *util_name*.

*User Response:* Add the missing DD statement.

CTJUVBE  
*jcl_statement_number* ERROR: *string* is not allowed for DD statement *dd_name* by *util_name*

*Explanation:* The message is issued when Control-M/JCL Verify detects that DD statement *jcl_statement_number* is a DUMMY, DYNAM, SYSOUT=, SUBSYS=, or DSN=NULLFILE, which are not allowed by the utility program.

*User Response:* Correct the DD statement.

CTJUVCE  
*jcl_statement_number* ERROR: DISP=disp is not allowed for DD statement *dd_name* by *util_name*

*Explanation:* The message is issued when Control-M/JCL Verify detects that DD statement *jcl_statement_number* has DISP=NEW or DISP=MOD, which are not allowed by the utility program.

*User Response:* Correct the DD statement.

CTJUVDI  
INFORMATION: *util_name* found *number1* error and *number2* warning messages

*Explanation:* The message is issued when Control-M/JCL Verify summarizes the number of error and warning messages found by the utility program.

*User Response:* Correct the errors.

CTJUVEE  
*util_name* ERROR: Utility validation for current member is interrupted: time-out is reached

*Explanation:* The message is issued when Control-M/JCL Verify, running in the IOA Online Monitor, detects that another user of the utility validation, such as SORT, is holding the resource for more than 10 seconds.

*User Response:* Try to run again.

CTJUVFE  
*util_name* ERROR: DD *dd_name_1* and *dd_name_2* are missing or DUMMY. One of them is required for utility

*Explanation:* The message is issued when Control-M/JCL Verify detects that the DD statements are missing, DUMMY, or NULLFILE.

*User Response:* Add or correct the DD statements.
CTJUVGW  \textit{jcl\_statement\_number} WARNING: Data set cannot be accessed by \textit{userid} - Utility \textit{util\_name} cannot be validated

\textit{Explanation:} The message is issued when Control-M/JCL Verify detects that the user performing the validation cannot access the file in the DD statement \textit{jcl\_statement\_number}.

\textit{User Response:} Contact the security administrator to get the required authorization for accessing the file.

CTJUVHE  \textit{jcl\_statement\_number} ERROR: member \textit{member\_name} does not exist. Utility \textit{util\_name} cannot be validated

\textit{Explanation:} The message is issued when Control-M/JCL Verify detects that the member in the DD statement \textit{jcl\_statement\_number} does not exist.

\textit{User Response:} Correct the DD statement.

CTJUVJE \textit{util\_name} ERROR: Invalid keyword \textit{string}

\textit{Explanation:} The message is issued when Control-M/JCL Verify detects that the keyword \textit{keyword} is not valid.

\textit{User Response:} Correct the keyword.

CTJUVKI \textit{util\_name} INFORMATION: duplicated keyword \textit{keyword} - keyword is ignored

\textit{Explanation:} The message is issued when Control-M/JCL Verify detects that keyword \textit{keyword} appears more than once in the statement or parameter list of the utility. The utility uses the first keyword and ignores all its duplications.

\textit{User Response:} Remove all the unnecessary keywords.

CTJUVLE \textit{util\_name} ERROR: Invalid data in keyword \textit{string}

\textit{Explanation:} The message is issued when Control-M/JCL Verify detects that keyword \textit{keyword} has invalid data.

\textit{User Response:} Correct the data in the keyword.

CTJUVNE \textit{util\_name} ERROR: The above \textit{string} statement is not valid.

\textit{Explanation:} The message is issued when Control-M/JCL Verify detects that the statement above contains an invalid keyword.

\textit{User Response:} Correct the statement using a valid keyword.
CTJUVOE  
*util_name*  ERROR: A control statement COPY, COPYGRP, COPYMOD or ALTERMOD must be before the above statement

*Explanation:* The message is issued when Control-M/JCL Verify detects that statement above (INDD, SELECT or EXCLUDE) requires a copy or ALTERMOD statement before it.

*User Response:* Add the missing control statement or remove the existing statement above.

CTJUVPI  
*util_name*  INFORMATION: Using PARM= with a copy option. SYSIN is ignored.

*Explanation:* The message is issued when Control-M/JCL Verify detects that PARM= in the EXEC JCL statement contains the copy option; therefore the SYSIN statement is ignored.

*User Response:* No action is required.

CTJUVQI  
*util_name*  INFORMATION: DD statement in the SYSIN concatenation or one of the SYSIN files is a DUMMY DD statement or DSN=NULLFILE. All the following DD statements are ignored.

*Explanation:* The message is issued when Control-M/JCL Verify detects that one of the DD statements in the SYSIN concatenation or in one of the SYSIN files is a DUMMY or DSN=NULLFILE. All the following DD statements are ignored.

*User Response:* No action is required.

CTJUVRE  
*util_name*  ERROR: Unknown Control statement *statement* - Statement is ignored

*Explanation:* The message is issued when Control-M/JCL Verify detects an unknown control statement.

*User Response:* Correct the DD statement.

CTJUVSE  
*util_name*  ERROR: Missing required DD *DD* for *command_type*

*Explanation:* The message is issued when Control-M/JCL Verify detects a *command_type* or operand that needs a *DD* statement.

*User Response:* Add the missing DD statement.

CTJUVTE  
*util_name*  ERROR: After standalone *command* command, the next COPY command is ignored

*Explanation:* The message is issued when Control-M/JCL Verify detects a standalone command, therefore the next COPY command is ignored.

*User Response:* No action is required.
CTJUVUE  **util_name** ERROR: After the previous statement, all standalone COPY type commands are ignored

*Explanation:* The message is issued when Control-M/JCL Verify detects that all the standalone COPY commands will be rejected, as a result of the previous control statement.

*User Response:* Remove the standalone commands.

CTJUVVE  **util_name** ERROR: Unknown Operand *Opr* - Statement is ignored

*Explanation:* The message is issued when Control-M/JCL Verify detects an unknown *Opr*.

*User Response:* Correct the operand statement.

CTJUVWE  **util_name** ERROR: Null Operand *Opr* - Statement is ignored

*Explanation:* The message is issued when Control-M/JCL Verify detects a NULL operand.

*User Response:* Correct the operand statement.

CTJUVXE  **util_name** ERROR: The above statement has an invalid continuation. Statement is ignored.

*Explanation:* The message is issued when Control-M/JCL Verify detects an invalid continuation, and therefore the statement is ignored.

*User Response:* Correct the continuation statement.

CTJUVYI  **util_name** Validate SYSIN type/num contains num records from src

*Explanation:* The message is issued as a heading summary of the SYSIN type, with the number of records, and the source of the records that Control-M/JCL Verify has checked.

*User Response:* No action is required.

CTJUVZ  **util_name** echo

*Explanation:* The message is issued as an echo of the statement checked by Control-M/JCL Verify. The suffix (s) of the message, E, I, or W, describes its severity.

Note: the suffixes of the message are:

- E - Error
- I - Information
- W - Warning.

*User Response:* If the message is CTJUVZI, no action is required. Otherwise perform the required process to correct the error or ignore the warning message.
CTJUW0I  *util_name* INFORMATION: IEBCOPY will use minimum block size of *minblk* and maximum block size of *maxblk*.

*Explanation:* The message is issued when Control-M/JCL Verify verifies a COPYMOD command that has MINBLK and MAXBLK. The message indicates the minimum and maximum block sizes that the COPYMOD will use during run time.

*User Response:* No action is required.

CTJUW1W  *util_name* WARNING: MAXBLK value is too small *maxblk*. IEBCOPY will use the default of 1024

*Explanation:* The message is issued when Control-M/JCL Verify detects that the COPYMOD command has a MAXBLK value that is less than 1024.

*User Response:* Correct the MAXBLK value.

CTJUW2W  *util_name* WARNING: MAXBLK value (*maxblk*) is above the limit. IEBCOPY will use the optimum maximum block size for the device.

*Explanation:* The message is issued when Control-M/JCL Verify detects that the COPYMOD command has a MAXBLK value that is higher than the maximum block size for the device.

*User Response:* Correct the MAXBLK value.

CTJUW3E  *util_name* ERROR: Operand *Opr* is not valid in statement statement.

*Explanation:* The message is issued when Control-M/JCL Verify detects that the operand in the command statement is not valid.

*User Response:* Remove the operand.

CTJUW4E  *util_name* ERROR: Statement EXCLUDE is not supported by COPYGRP

*Explanation:* The message is issued when Control-M/JCL Verify detects an EXCLUDE statement together with COPYGRP.

*User Response:* Remove the EXCLUDE statement.

CTJUW5I  *util_name* IEBCOPY will use minimum block size of *minblk* and the optimum maximum block size for the device.

*Explanation:* The message is issued when Control-M/JCL Verify verifies a COPYMOD command that has MINBLK. The message indicates the minimum and maximum block sizes that the COPYMOD will use during run time.

*User Response:* No action is required.
**CTJUW6**  *util_name WARNING: MAXBLK maxblk is lower than MINBLK minblk. MINBLK will be use as MAXBLK.*

*Explanation:* The message is issued when Control-M/JCL Verify verifies a COPYMOD command that has a MAXBLK value that is smaller than the MINBLK value.

*User Response:* Correct the MAXBLK value.

**CTJUW7**  *util_name WARNING: Dataset DD DD does not exist. DSN=dataset_name.*

*Explanation:* The message is issued when Control-M/JCL Verify detects that a required file, specified in a DD statement for the COPY or ALTERMOD command, does not exist.

*User Response:* Correct the file name.

**CTJUW8**  *jcl_statement_number ERROR: RECFM 'U' required with command for ddname ddname*

*Explanation:* The message is issued when Control-M/JCL Verify detects the use of DD with RECFM, different from 'U' with control statement ALTERMOD or COPYMOD.

*User Response:* Correct the DD statement.

**CTJUW9**  *util_name ERROR: Concatenation is not allowed DD ddname.*

*Explanation:* The message is issued when Control-M/JCL Verify detects concatenation while it is not allowed.

*User Response:* Delete the concatenation.

**CTJUWAE**  *util_name ERROR: DSORG 'PO' required with command for ddname ddname.*

*Explanation:* The message is issued when Control-M/JCL Verify detects that the DD is not a PDS as required when using command.

*User Response:* Correct the DD name.

**CTJUWBE**  *util_name ERROR: The above statement has unequal parentheses.*

*Explanation:* The message is issued when Control-M/JCL Verify detects that statement has unequal parentheses.

*User Response:* Correct the statement.

**CTJUWCE**  *util_name ERROR: No mixing of SELECT and EXCLUDE in same copy step*

*Explanation:* The message is issued when Control-M/JCL Verify detects that SELECT and EXCLUDE are mixed in same copy command.

*User Response:* Correct the statement.
CTJUWDW  util_name WARNING: Compress is ignored when copy step with SELECT

Explanation: The message is issued when Control-M/JCL Verify detects a compress command in a copy step, which contains SELECT.

User Response: Correct the statement.

CTJUWEE  util_name ERROR: RENAME and REPLACE are not allowed in EXCLUDE command

Explanation: The message is issued when Control-M/JCL Verify detects RENAME and REPLACE in an EXCLUDE command.

User Response: Correct the statement.

CTJUWF W  util_name WARNING: Found duplicate member name member_name - member will not be copied.

Explanation: The message is issued when Control-M/JCL Verify detects a duplicated member name member_name.

User Response: Remove the duplicated members.

CTJUWG W  util_name WARNING: Found old member name member_name and new member name are the same. Member will not be copied.

Explanation: The message is issued when Control-M/JCL Verify detects that the old member name and the new member name in the SELECT statement are the same. Therefore the member will not be copied.

User Response: Change the old or the new member name.

CTJUWHE  util_name ERROR: Invalid member name member_name

Explanation: The message is issued when Control-M/JCL Verify detects invalid member_name name.

User Response: Correct the member name.

CTJUWIE  util_name ERROR: Statement statement has no operand.

Explanation: The message is issued when Control-M/JCL Verify detects a statement without an operand.

User Response: Correct the statement.

CTJUWJE  util_name Duplicate keyword keyword

Explanation: The message is issued when Control-M/JCL Verify detects a duplicated keyword.

User Response: Remove the duplicated keyword.
CTJUWKE  

**util_name** ERROR: Missing mandatory keyword *keyword* in the above statement

*Explanation:* The message is issued when Control-M/JCL Verify detects that a mandatory keyword is missing in the above statement.

*User Response:* Correct the statement.

CTJUWLE  

**util_name** ERROR: INVALID REPLACE SPECIFIED - Statement is ignored

*Explanation:* The message is issued when Control-M/JCL Verify detects that a member contains an invalid REPLACE (R) operator.

*User Response:* Change the REPLACE operator to "R".

CTJUWME  

**util_name** ERROR: REPLACE is not allowed in *command*

*Explanation:* The message is issued when Control-M/JCL Verify detects that a member contains a REPLACE (R) operator as part of ALTERGRP or COPYGRP.

*User Response:* Remove the REPLACE operator.

CTJUWNE  

**util_name** ERROR: COPYMOD - INDD can not equal OUTDD. DDNAMEs are *dd_name*

*Explanation:* The message is issued when Control-M/JCL Verify detects that that in the COPYMOD statement, the INDD and OUTDD are the same DD name.

*User Response:* Change the INDD or OUTDD.

CTJUWOE  

**util_name** ERROR: COPYMOD - INDD in_dd_name and OUTDD out_dd_name have the same DSN=dataset_name

*Explanation:* The message is issued when Control-M/JCL Verify detects that in the COPYMOD statement, the INDD and OUTDD use the same dataset.

*User Response:* Change the DSN (dataset name) INDD or OUTDD.

CTJUWQE  

jcl_statement_number ERROR: **util_name** – *dd_name* file is PDS or PDSE but member is missing. DSN=dataset_name

*Explanation:* The message is issued when Control-M/JCL Verify detects that the file is a PDS or PDSE but no member name was specified in the DSN=dataset_name or the member related statement or keyword is missing in the util_name.

*User Response:* Add the member name in the dataset_name or the util_name statement.
CTJUWRE  jcl_statement_number  ERROR: command - ddname RECFM recfm is not valid or missing.

Explanation: The message is issued when Control-M/JCL Verify either detects that the ddname's RECFM is not valid for this utility, or that the file is a DUMMY file and the RECFM is missing.

User Response: Either correct the DSN so that it has a valid RECFM, or for a DUMMY file, ensure that it has a valid RECFM.

CTJUWSE  util_name  ERROR: RECFM INCOMPATIBLE - INDD in_dd_name RECFM recfm OUTDD out_dd_name RECFM recfm

Explanation: The message is issued when Control-M/JCL Verify detects that the INDD and OUTDD attributes are incompatible.

User Response: Correct the files attributes.

CTJUWTE  jcl_statement_number  ERROR: RECORD length value inconsistent with BLOCK size value for RECFM=record_format

Explanation: The message is issued when Control-M/JCL Verify detects that the file jcl_statement_number LRECL value and the BLKSIZE value are incompatible with the record format’s rule. This can be lead to I/O errors (ABEND) when the utility writes to the file.

The values for LRECL, BLKSIZE, or RECFM can be specified in the JCL statement or copied from the INDD file.

User Response: Correct the files attributes.

CTJUWE  util_name  ERROR: Multiple INDD invalid for command

Explanation: The message is issued when Control-M/JCL Verify detects that there are multiple INDDs, which are invalid for the same command.

User Response: Remove the multiple INDDs, or change them so each INDD is part of a different COPYGRP or COPYGROUP statement.

CTJUWVI  util_name  INFORMATION: operand is not a PDS. DD=ddname DSN=dataset_name

Explanation: The message is issued when Control-M/JCL Verify detects that the file is not a PDS or PDSE. If the file is an INDD, IEBCOPY performs a restore. Otherwise IEBCOPY performs a backup.

Note: IEBCOPY might ignore the DCB parameters in the JCL or file, and instead determine the attributes using the process and file contents.

User Response: No action is required.
CTJUWWI  **util_name** INFORMATION: OUTPUT DATASET RECFM/LRECL/BLKSIZE
COPIED FROM INPUT DATASET

*Explanation:* The message is issued when Control-M/JCL Verify detects that the DCB attributes of the OUTPUT file are not all set, and IEBCOPY will complete the attributes from the first input file.

*User Response:* If the DCB attributes are not specified as required, set the RECFM, LRECL, or BLKSIZE values in the JCL definition of the OUTPUT file.

CTJUXI  **jcl_statement_number** INFORMATION: command for file type type is not relevant

*Explanation:* The message is issued when Control-M/JCL Verify detects that *command* is for a file type that is not relevant.

*User Response:* No action is required.

CTJUYE  **util_name** ERROR: Invalid new member name *string*

*Explanation:* The message is issued when Control-M/JCL Verify detects that the new member name *string* in the MEMBER operand is invalid.

*User Response:* User response: Correct the member name.

CTJUZI  **util_name** INFORMATION: File Access (FA), File Existence (FE) or Module Existence (ME) disabled. Utility cannot be verified.

*Explanation:* The message is issued when Control-M/JCL Verify detects that one or more of the following functions, File Access (FA), File Existence (FE), and Module Existence (ME), which are required for the utility validation, are not active. Therefore, the utility cannot be validated.

*User Response:* Ensure that the IOA Security is installed and active, and then set the parameters as follows: FA=Y, FE=Y, and ME=Y.

CTJUX3W  **util_name** WARNING: operand request *number-1* using *number-2*

*Explanation:* The message is issued when Control-M/JCL Verify verifies a COPYMOD command that has a MAXBLK or MINBLK, which is not set according to the assignment in the statement.

The reasons might be:

- MAXBLK is lower than the MINBLK. The MINBLK set as the MAXBLK.
- The file BLOCK SIZE is lower than the MAXBLK. MAXBLK is set to the file block size.
- The OUTDD library BLOCK SIZE is zero. IEBCOPY uses the INDD BLOCK SIZE.
Note: If Control-M/JCL Verify cannot determine the OUTDD BLOCK SIZE, the maximum value allowed is 32760.

User Response: Adjust the MAXBLK or MINBLK values.

CTJUX4I  
util_name INFORMATION: operand value number1 is above the limit. IEBCOPY will use number2

Explanation: The message is issued when Control-M/JCL Verify detects COPYMOD command has operand (MAXBLK or MINBLK) that is higher than maximum block allowed (32760).

User Response: Correct the MAXBLK value.

CTJUX5I  
util_name INFORMATION: operand value number1 is below the limit. IEBCOPY will use number2

Explanation: The message is issued when Control-M/JCL Verify detects COPYMOD command has operand (MAXBLK or MINBLK) that is lower than minimum block size allowed (1024).

User Response: Correct the MINBLK value.

CTJUX6E  
util_name ERROR: DD ddname DSN is a PDS with member, but this member does not exist.

Explanation: The message is issued when Control-M/JCL Verify detects that the utility refers to a DD with a DSN referencing a PDS or PDS/E with the member member. However, the member member does not exist, and as result, the utility might fail because of an OPEN error.

CTJV00E  
Rule=<rule> Table=<table> Command=<cmd> RC=<return_code> RS=<reason_code>

Explanation: DO <cmd> in the given rule failed due to a syntax error.

User Response: See following message CTJV03I which explains the error and correct the DO command accordingly.

CTJV01I  
Text=<text>

Explanation: The message shows the text of the command that failed.

User Response: No action is required.

CTJV02S  
Rule=<rule> Table=<table> Command=<cmd> RC=<return_code> RS=<reason_code>

Explanation: Command <cmd> in Rule <rule> Table <table> failed with RC <return_code> RS <reason_code>.

User Response: Contact BMC support.
CTJ03I  <explanation>

Explanation: The message shows the text of the command that failed.
User Response: No action is required.

CTMD78I  JCL VERIFICATION CAN ONLY BE DONE FOR JOBS

Explanation: The message explains the error.
User Response: No action is required.

CTMD79I  JCL VERIFICATION CAN'T BE DONE FOR ON-POOL JOBS

Explanation: Control-M/JCL Verify cannot verify on-spool jobs.
User Response: No action is required.

CTMD7AI  JCL VERIFY OK, RERUN OF job_name TABLE table_name ODATE odate PERFORMED

Explanation: JCL verification was performed while rerunning a job. The JCL verification finished OK, and the rerun was successful.
User Response: No action is required.

CTMD7BI  JOB NOT RERUN DUE TO JCL VERIFICATION FAILURE

Explanation: JCL verification was performed before rerunning a job. The JCL did not successfully pass the verification, and therefore the job was not rerun.
User Response: No action is required.

CTMD7CI  JCL VERIFICATION CAN'T BE DONE FOR DUMMY JOBS

Explanation: Control-M/JCL Verify cannot verify dummy jobs.
User Response: No action is required.

CTMD7E1  JCL VERIFICATION CAN'T BE DONE FOR WARNING JOBS

Explanation: Control-M/JCL Verify cannot verify warning jobs.
User Response: No action is required.

CTMD7FW  CONTROL-M JCL VERIFY IS NOT INSTALLED

Explanation: Control-M/JCL Verify is not installed.
User Response: No action is required.
CTMD7GI  JOB HAS CHANGED. INVOKE JVER AFTER RE-ENTERING THIS SCREEN

Explanation: This message is issued in screen 2 and screen 3. If the job scheduling definition is changed, and then the JVER command is issued, this message is displayed.

User Response: Exit screen 2 or screen 3, and re-enter.

IOAJV0E  INVALID VALUE, USE "Y", "N" OR "D"

Explanation: An invalid value was specified for a verification criteria. The cursor points to the field that contains the invalid value.

User Response: Correct invalid value.

IOAJV1E  INVALID VALUE, USE "I", "W", "E" OR "D"

Explanation: An invalid value was specified for a verification criteria. The cursor points to the field that contains the invalid value.

User Response: Correct invalid value.

IOAJV2E  INVALID OPTION FOR SMART TABLE

Explanation: “E” (JCL VERIFY) option was entered for table entity.

User Response: Use valid option.

IOAJV3E  INVALID VALUE. RULE CAN CONTAIN ONE TYPE OF "ON"

Explanation: Related ON statement has more than one type.

User Response: Delete invalid ON statement, by filling in a valid type or setting blank the O/N option of the previous ON statement.

IOAJV4E  INVALID VALUE, USE "O", "N" OR BLANK

Explanation: Invalid O/N option was specified. The valid options are O for an OR relationship, N for a NOT relationship, or blank for no additional ON line.

User Response: Fill in O, N or leave blank.

IOAJV5E  INVALID NESTING OF IF-ELSE-ENDIF.PLEASE CHECK

Explanation: Rule definition contains invalid nesting of DO IF, DO ELSE and DO ENDIF statements.

User Response: Correct nesting.
Sample exit 4

Control-M/JCL Verify is delivered with a CA Endevor sample exit 4 (CTJAPIA) that uses CTJAPI. This exit is in the IOA.SAMPLE library. Whenever an element is checked in, exit 4 is invoked.

The CA Endevor sample (using exit 4) performs the following:

- identifies that a JCL job element is checked in
- verifies that the job definition is valid
- fails the check-in, if any errors are found
Editing Rule Definitions in the IOA Edit Environment

Rule Definition parameters can be edited (moved, copied, deleted, repeated) by performing IOA Line Editing commands similar to standard ISPF line commands, from within the IOA Edit environment.

The Edit environment in a Rule Definition screen is accessed by typing EDIT in the COMMAND field and pressing Enter.

Figure 75  Edit Environment in a Rule Definition Screen

| - CONTROL-M/JCLVERIFY  RULE:  DOTYP05 | TABLE: DOTYPES0 |
| COMMAND ====> | SCROLL====> CRSR |
| | +-----------------------------------------------------------------------------+ |
| | __ RULE DOTYP05     ENVIRONMENT GENERAL     PRIORITY 50    CONTINUE SEARCH Y |
| | __ DESC DOTYP05 RETURN |
| | __ DESC |
| | __ =========================================================================== |
| | __ JOBNAME * |
| | __ LIBRARY * |
| | __ SCHDLIB |
| | __ =========================================================================== |
| | __ ON JOB           CLASS         PRTY          USER |
| | | ACCOUNT |
| | | O/N |
| | | =========================================================================== |
| | __ DO SET         %%Q = %%$BLANK |
| | __ DO |
| | __ IF         %%Q EQ %%$BLANK |
| | __ DO MSG         TYPE E RETURN ACTIVATED RC 0010 RS 2222 |
| | __ DO |
| | __ RETURN   RC 0010 RS X' 2222 ' |
| | __ ENDIF |
| | __ DO SET         %%A = RULE DOTYP06 %% PLUS |
| | __ DO SET         %%B = PR 50 CONTINUE SEARCH Y |
| | __ DO MSG         TYPE I TEST FOR %%A %%B |
| | __ DO SET         %%E = %%C%%$PLUS%%D |
| | __ DO SET         %%G = 1 |
| | __ DO SET         %%D = 2 |
| | __ DO MSG         TYPE I %%C PLUS %%D IS %%E |
| | __ DO |
| | FILL IN RULE DEFINITION. CMDS: EDIT DOOPT ONOPT CANCEL 18.52.23 |
A 2-character line editing command field, marked by underscores, is displayed for each line on the Rule Definition screen.

Editing commands are typed directly onto these underscores.

Incorrectly specified line editing commands can be corrected by typing over them correctly. Line editing commands can be deleted by blanking them out.

Specified line editing commands are processed when Enter is pressed.

Control-M/JCL Verify performs automatic syntax checking to ensure that the rule definition is still syntactically correct after editing. If an edit will invalidate the rule definition, a message is displayed at the top of the screen and the edit is not performed. For guidelines and recommendations for editing rule definitions see “Maintaining Valid Rule Definitions” on page 238.

All operations available in the Rule Definition screen can be performed while in the Edit environment. For example, parameter values can be changed, the Rule Definition screen can be saved, and the screen exited.

To exit the Edit environment, retype EDIT in the COMMAND field and press Enter. Line editing command fields are removed from the display.

Line editing commands can be performed on any single ON or DO statement or on a block of ON or DO statements.

### Line Editing Commands

The following tables show the types of line editing commands that exist in the Edit environment.

#### Table 48  Delete Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS</td>
<td>Delete a single line.</td>
</tr>
<tr>
<td>DL</td>
<td>Delete a logical line.</td>
</tr>
<tr>
<td>DD</td>
<td>Delete lines between two DD specifications.</td>
</tr>
<tr>
<td>D</td>
<td>Deletes a line. Control-M/JCL Verify determines whether to delete a single or logical line based on the parameter on the line.</td>
</tr>
</tbody>
</table>
### Table 49  Copy commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS</td>
<td>Copies a single line.</td>
</tr>
<tr>
<td>CL</td>
<td>Copies a logical line.</td>
</tr>
<tr>
<td>CC</td>
<td>Copies lines between two CC specifications.</td>
</tr>
<tr>
<td>C</td>
<td>Copies a line. Control-M/JCL Verify determines whether to copy a single or logical line based on the parameter on the line.</td>
</tr>
</tbody>
</table>

Copy commands are used in conjunction with Location commands. The lines are placed at the position indicated by the A or B Location command (described in Table 53).

### Table 50  Move commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS</td>
<td>Move a single line.</td>
</tr>
<tr>
<td>ML</td>
<td>Moves a logical line.</td>
</tr>
<tr>
<td>MM</td>
<td>Moves lines between two MM specifications.</td>
</tr>
<tr>
<td>M</td>
<td>Moves a line. Control-M/JCL Verify determines whether to move a single or logical line based on the parameter on the line.</td>
</tr>
</tbody>
</table>

Move commands are used in conjunction with Location commands. The lines are placed at the position indicated by the A or B Location command (described in Table 53).

### Table 51  Repeat commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS</td>
<td>Repeat a single line.</td>
</tr>
<tr>
<td>RL</td>
<td>Repeats a logical line.</td>
</tr>
<tr>
<td>RR</td>
<td>Repeats lines between two RR specifications.</td>
</tr>
<tr>
<td>R</td>
<td>Repeats a line. Control-M/JCL Verify determines whether to repeat a single or logical line based on the parameter on the line. The repeated lines is placed immediately after the lines that has been repeated.</td>
</tr>
</tbody>
</table>

### Table 52  Insert commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I (Insert)</td>
<td>Insert a new logical line or block after the logical line or block marked with an I.</td>
</tr>
</tbody>
</table>
Maintaining Valid Rule Definitions

Since rule definitions must be syntactically correct at all times, the user must consider the following issues when specifying Line Editing commands:

- The result of a line editing command is dependent on the line on which the command is specified. For example, the D command deletes either a single or a logical line based on the line type.

- Logical lines function as a unit and must not be separated.

  When a logical command is specified within a logical line, that is, on a subparameter line, or a continuation line, the specified operation is performed on the entire logical line.

- Blank parameter lines added automatically by Control-O, to allow the user to specify additional parameters, cannot be deleted.

It is recommended that, wherever possible, the D, C, R, and M commands be used for editing (instead of DS, DL, CS, CL, RS, RL, MS, and ML) because these commands automatically retain the logical structure of the rule definition.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (After)</td>
<td>Indicates that lines should be placed after the line marked with an A.</td>
</tr>
<tr>
<td>B (Before)</td>
<td>Indicates that lines should be placed before the line marked with a B.</td>
</tr>
</tbody>
</table>

The A and B Location commands are used together with Copy (C, CS, CL, CC), and Move (M, MS, ML, MM) commands.
Figure 76  Example 1 Before – Move a DO Block in the Rule Definition Screen

- CONTROL-M/JCLVERIFY  RULE:  DOTYP05  TABLE:  DDTYPES0
COMAND ===>
SCROLL==> CRSR
+
- RULE DOTYP05  ENVIRONMENT GENERAL  PRIORITY 50  CONTINUE SEARCH Y
- DESC DOTYP05 RETURN
- DESC

==============================================================================

JOBNAME *
==============================================================================

LIBRARY *  MEMBER  DOTYPE01
SCHDLIB  SCHDTAB  O/N

==============================================================================

ON JOB  CLASS  PRTY  USER
ACCOUNT  O/N

==============================================================================

DO SET  %Q = %$BLANK
DO
  IF  %Q EQ %$BLANK
  DO MSG  TYPE E RETURN ACTIVATED RC 0010 RS 2222
  DO
  RETURN  RC 0010 RS X' 2222 ' 
  ENDF
  DO SET  %A = RULE DOTYP06 % PLUS
  DO SET  %B = PR 50 CONTINUE SEARCH Y
  DO MSG  TYPE I TEST FOR %A %B
  M_ DO SET  %E = %C%%PLUS%%D
  DO SET  %C = 1
  A_ DO SET  %D = 2
  DO MSG  TYPE I %C PLUS %D IS %E
  DO

FILL IN RULE DEFINITION. CMDS: EDIT  DOOPT  ONOPT  CANCEL  18.52.23
Figure 77  Example 1 After – Move a DO Block in the Rule Definition Screen (the DO block has been moved)

```plaintext
- CONTROL-M/JCLVERIFY  RULE:  DOTYP05  TABLE:  DOTYPES0
COMMAND ===>  SCROLL ===>  CRSR
+-----------------------------------------------------------------------------+
| RULE DOTYP05     ENVIRONMENT GENERAL     PRIORITY 50    CONTINUE SEARCH Y |
| DESC DOTYP05 RETURN                                                      |
| DESC                                                             |
+-----------------------------------------------------------------------------+
| __ JOBNAME *                                                          |
+-----------------------------------------------------------------------------+
| __ LIBRARY *    MEMBER DOTYPE01                                        |
| SCHEDLIB  SCHDTAB  O/N                                                 |
+-----------------------------------------------------------------------------+
| __ ON JOB     CLASS          PRTY         USER      ACCOUNT |
|                                                        O/N |
+-----------------------------------------------------------------------------+
| __ DO SET     %%Q = %%$BLANK                                            |
| __ DO                                                  |
| __ IF       %%Q EQ %%$BLANK                                            |
| __ DO MSG     TYPE E RETURN ACTIVATED RC 0010 RS 2222                   |
| __ DO                                                  |
| __ RETURN   RC 0010 RS X' 2222 '                                 |
| __ ENDF                                                |
| __ DO SET     %%A = RULE DOTYP06 %% PLUS                                |
| __ DO SET     %%B = PR 50 CONTINUE SEARCH Y                             |
| __ DO MSG     TYPE I TEST FOR %%A %%B                                 |
| __ DO SET     %%C = 1                                                  |
| __ DO SET     %%D = 2                                                  |
| __ DO SET     %%E = %%C%%$PLUS%%D                                      |
| __ DO MSG     TYPE I %%C PLUS %%D IS %%E                               |
| __ DO                                                  |
+-----------------------------------------------------------------------------+
FILL IN RULE DEFINITION. CMDS: EDIT DOOPT ONOPT CANCEL 18.52.23
```
### Figure 78 Example 2 Before – Copy a Line of Text in the Rule Description

```
- CONTROL-M/JCLVERIFY RULE: DDTYPE05
COMMAND ===> SCROLL====> CRSR
+-----------------------------------------------------------------------------+
__ RULE DDTYPE05     ENVIRONMENT GENERAL     PRIORITY 50    CONTINUE SEARCH Y
C_ DESC DDTYPE05 RETURN
A_ DESC

VALUES
=================================

JOBNAME *

LIBRARY *
SCHDLIB SCHDTAB O/N

ON JOB CLASS PRTY USER
ACCOUNT O/N

DO SET %%Q = %%$BLANK
DO
IF %%Q EQ %%$BLANK
DO MSG TYPE E RETURN ACTIVATED RC 0010 RS 2222
DO
RETURN RC 0010 RS X' 2222 '
ENDIF
DO SET %%A = RULE DDTYPE06 %% PLUS
DO SET %%B = PR 50 CONTINUE SEARCH Y
DO MSG TYPE I TEST FOR %%A %%B
DO SET %%C = 1
DO SET %%D = 2
DO SET %%E = %%C%%$PLUS%%D
DO MSG TYPE I %%C PLUS %%D IS %%E
DO
FILL IN RULE DEFINITION. CMDS: EDIT DOOPT ONOPT CANCEL 18.52.23
```
**Figure 79**  Example 2 After – Copy a Line of Text in the Rule Description (the text has been copied)

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>SCROLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTROL-M/JCLVERIFY</td>
<td>CRSR</td>
</tr>
</tbody>
</table>

- RULE DOTYPES  
  ENVIRONMENT GENERAL  
  PRIORITY 50  
  CONTINUE SEARCH Y  
- DESC DOTYPES RETURN  
- DESC  
- DESC DOTYPES RETURN  
- JOBNAME *  
- LIBRARY *  
  MEMBER DOTYPE01  
  SCHDLIB SCHDTAB O/N  
- ON JOB CLASS PRTY USER  
- ACCOUNT O/N  
- DO SET %Q = %$BLANK  
- DO IF %Q EQ %$BLANK  
- DO MSG TYPE E RETURN ACTIVATED RC 0010 RS 2222  
- DO RETURN RC 0010 RS X'2222'  
- ENDIF  
- DO SET %A = RULE DOTYPE06 % PLUS  
- DO SET %B = PR 50 CONTINUE SEARCH Y  
- DO MSG TYPE I TEST FOR %A %B  
- DO SET %C = 1  
- DO SET %D = 2  
- DO SET %E = %C%PLUS%D  
- DO MSG TYPE I %C PLUS %D IS %E  

FILL IN RULE DEFINITION. CMDS: EDIT DOOPT ONOPT CANCEL 18.52.23
Figure 80  Example 3 Before – Delete an Unnecessary DO Statement

```
- CONTROL-M/JCLVERIFY RULE: DOTYP05  TABLE: DOTYPES0
COMMAND ===>                                                    SCROLL==> CRSR
+-----------------------------------------------------------------------------+
__ RULE DOTYP05     ENVIRONMENT GENERAL     PRIORITY 50    CONTINUE SEARCH Y
__ DESC DOTYP05 RETURN
__ DESC
__ DESC DOTYP05 RETURN
  ===========================================================================
  JOBNAME *
  ===========================================================================
  LIBRARY *                      MEMBER DOTYPE01
  SCHDLIB                          SCHDTAB O/N
  ===========================================================================
  ON JOB     CLASS     PRTY     USER
  ACCOUNT                                                                 O/N
  ===========================================================================
  DO SET      %%Q = %%$BLANK
  DO
  IF       %%Q EQ %%$BLANK
  DO MSG      TYPE E RETURN ACTIVATED RC 0010 RS 2222
  DO
  RETURN   RC 0010 RS X' 2222 '
  ENDIF
  __ DO SET      %%A = RULE DOTYP06 %% PLUS
  __ DO SET      %%B = PR 50 CONTINUE SEARCH Y
  __ DO MSG      TYPE I TEST FOR %%A %%B
  __ DO SET      %%C = 1
  __ DO SET      %%D = 2
  __ DO SET      %%E = %%C%%$PLUS%%D
  D_ DO MSG      TYPE I %%C PLUS %%D IS %%E
  __ DO

FILL IN RULE DEFINITION, CMDS: EDIT DOOPT ONOPT CANCEL
```
Figure 81  Example 3 After – Delete an Unnecessary DO Statement (the DO MSG statement has been deleted)

- CONTROL-M/JCLVERIFY  RULE:  DOTYP05  TABLE:  DOTYPES0
  COMMAND ===>  SCROLL===>  CRSR
  
  __ RULE DOTYP05  ENVIRONMENT GENERAL  PRIORITY 50  CONTINUE SEARCH Y
  __ DESC DOTYP05 RETURN
  __ DESC
  __ DESC DOTYP05 RETURN
  __ ===========================================================================
  __ JOBNAME *
  __ ===========================================================================
  __ LIBRARY *  MEMBER  DOTYPE01
  __ SCHDLIB  SCHDTAB  O/N
  __ ===========================================================================
  __ ON JOB  CLASS  PRTY  USER
  __ ACCOUNT  O/N
  __ ===========================================================================
  __ DO SET  %%Q = %%$BLANK
  __ DO
  __ IF  %%Q EQ %%$BLANK
  __ DO MSG  TYPE E RETURN ACTIVATED RC 0010 RS 2222
  __ DO
  __ RETURN  RC 0010 RS X' 2222 '
  __ ENDIF
  __ DO SET  %%A = RULE DOTYP06 %% PLUS
  __ DO SET  %%B = PR 50 CONTINUE SEARCH Y
  __ DO MSG  TYPE I TEST FOR %%A %%B
  __ DO SET  %%C = 1
  __ DO SET  %%D = 2
  __ DO SET  %%E = %%C%%$PLUS%%D
  __ DO
  __ FILL IN RULE DEFINITION. CMDS: EDIT DOOPT ONOPT CANCEL 18.52.23
Figure 82  Example 4 Before – Move specified DO statements in the Rule Definition

```
- CONTROL-M/JCLVERIFY  RULE: DOTYP05                           TABLE: DOTYPES0
COMMAND ===>                                                    SCROLL==> CRSR
+-----------------------------------------------------------------------------+
__ RULE DOTYP05     ENVIRONMENT GENERAL     PRIORITY 50    CONTINUE SEARCH Y
__ DESC DOTYP05 RETURN
__ DESC
__ DESC DOTYP05 RETURN
__ ===========================================================================
__ JOBNAME *
__ ===========================================================================
__ LIBRARY *                               MEMBER  DOTYPE01
__ SCHDLIB                                  SCHDTAB          O/N
__ ===========================================================================
__ ON JOB CLASS PRTY USER
__ ACCOUNT                                                                 O/N
__ ===========================================================================
MM DO SET      %%Q = %%$BLANK
__ DO
__ IF       %%Q EQ %%$BLANK
__ DO MSG      TYPE E RETURN ACTIVATED RC 0010 RS 2222
__ DO
__ RETURN   RC 0010 RS X' 2222 ' 
MM ENDIF
__ DO SET      %%A = RULE DOTYP06 %% PLUS
__ DO SET      %%B = PR 50 CONTINUE SEARCH Y
__ DO MSG      TYPE I TEST FOR %%A %%B
__ DO SET      %%C = 1
__ DO SET      %%D = 2
A_ DO SET      %%E = %%%C%%PLUS%%D
__ DO
FILL IN RULE DEFINITION. CMDS: EDIT DOOPT ONOPT CANCEL 18.52.23
```
Figure 83  Example 4 After – Move specified DO statements in the Rule Definition  
(the DO statements Have Been Moved)

```
- CONTROL-M/JCLVERIFY  RULE:  DOTYP05  
  COMMAND ===>  
  +-----------------------------------------------------------------------------+
  _ RULE DOTYP05     ENVIRONMENT GENERAL     PRIORITY 50    CONTINUE SEARCH Y 
  _ DESC DOTYP05 RETURN 
  _ DESC 
  _ DESC DOTYP05 RETURN 
  _==================================================================
  _ JOBNAME * 
  _==================================================================
  _ LIBRARY * 
  _ SCHDLIB 
  _ MEMBER DOTYPE01 
  _ SCHDTAB O/N   
  _==================================================================
  _ ON JOB     CLASS     PRTY     USER     
  _ ACCOUNT O/N   
  _==================================================================
  _ DO SET      %%A = RULE DOTYP06 %% PLUS 
  _ DO SET      %%B = PR 50 CONTINUE SEARCH Y 
  _ DO MSG     TYPE I TEST FOR  %%A  %%B 
  _ DO SET      %%C = 1 
  _ DO SET      %%D = 2 
  _ DO SET     %%E = %%C%%$PLUS%%D 
  _ DO SET     %%Q = %%$BLANK 
  _ DO 
  _ IF      %%Q EQ %%$BLANK 
  _ DO MSG     TYPE E RETURN ACTIVATED RC 0010 RS 2222 
  _ DO 
  _ RETURN   RC 0010 RS X' 2222 ' 
  _ ENDF 
  _ DO 
  _==================================================================
  _ FILL IN RULE DEFINITION. CMDS: EDIT DOOPT ONOPT CANCEL 18.52.23
```
The %%$PARSE function

Overview of %%$PARSE ................................................................. 247
Parsing Words ................................................................. 248
Using Dummy Variables (Place Holders) ......................... 249
Using Patterns in Parsing .................................................. 250
  Using String Patterns .................................................... 250
  Using Numeric Patterns Within the Template ................. 252
  Using More Than One Pattern and Combining Pattern Types in the Template 254

Overview of %%$PARSE

The %%$PARSE function is a powerful tool that offers extensive string manipulation capabilities in Control-M Workload Automation JCL Verify environment. This function, which is similar to the REXX PARSE command in the TSO/E environment, can be used to analyze and extract information from various variables, such as System variables, card data variables and user defined variables.

The %%$PARSE function parses a specified string (that is, it splits the specified string into substrings) according to a specified template. A template consists of variables and “patterns” that determine the parsing process.

The format of the %%$PARSE function is

DO SET=%%$PARSE string template

In this format

- `string` is the AutoEdit variable that contains the string to be parsed
- `template` is the AutoEdit variable or constant that contains the template
Example

```
DO SET=%%S=THIS IS A SAMPLE STRING
DO SET=%%T=A1 A2 A3 A4 A5
DO SET=%%$PARSE %%S %%T
```

The % %$PARSE function assigns substrings of the specified string to the specified variables according to the specified template.

The DO SET statements in the above example provide the same result as the following DO SET statements:

```
DO SET=%%A1=THIS
DO SET=%%A2=IS
DO SET=%%A3=A
DO SET=%%A4=SAMPLE
DO SET=%%A5=STRING
```

The parsing process involves the following stages:

1. The string is broken into substrings, from left to right, using the patterns in the template.

2. Each substring is parsed into words, from left to right, using the variable names in the template.

Template elements are

- String Patterns
- Position Patterns
- Variables
- Place holders (Dummy variables)

The rules of parsing are detailed in the following paragraphs.

**Parsing Words**

Scanning is performed from left to right and words in the string (leading and trailing blanks excluded) are matched one by one with the variables named in the template. The last variable named in the template will contain the remaining part of the string, including leading and trailing blanks.

Up to 30 variable names can be specified in a parsing template.
The following situations can be encountered:

- The number of words in the string matches the number of variables in the template.
  Each of those variables contains one word of the string. The last variable contains the last word in the string including leading and trailing blanks.

- The number of words in the string is smaller than the number of variables named in the template
  The first variables each contain one word of the string and the extra variables receive a value of NULL (a string of 0 character length).

- The number of words in the string is greater than the number of variables in the template
  All variables but the last one contain one word of the string and the last variable named in the template contains the remaining part of the string, including leading and trailing blanks.

**Example**

The DO SET statements below (which include a `%%$PARSE` function)

```
DO SET=%%S = THIS IS A SAMPLE STRING
DO SET=%%T = A1 A2 A3
DO SET=%%$PARSE %%S %%T
```

have the same result as the following DO SET statements:

```
DO SET=%%A1 = THIS
DO SET=%%A2 = IS
DO SET=%%A3 = A SAMPLE STRING
```

**Using Dummy Variables (Place Holders)**

A single period can be used as a dummy variable in the template. This is useful when the corresponding word in the string does not need to be stored in a named variable.

**Example**

The following DO SET statements (which include a `%%$PARSE` function)

```
DO SET=%%S = THIS IS A SAMPLE STRING
DO SET=%%$PARSE %%S %%T
```
Using Patterns in Parsing

Patterns can be included in the template. Their purpose is to break down the string into substrings prior to the actual parsing into words process. Parsing will then be performed, as previously described, on the substrings and not on the original string.

Two types of patterns are available:

- **String Patterns** – a character string delimited by quotes, to distinguish it from a variable name
- **Numeric (Positional) Patterns** – a number, signed or unsigned

Using String Patterns

The string is scanned from left to right for a substring that matches the string pattern.

The following situations may occur:

- A match is found, that is, a substring within the string is identical to the given string pattern.

The original string is divided into two substrings. The first substring (up to, but not including, the string pattern) is parsed into words using the variables named before the string pattern on the template. Parsing continues from the character following the matched string.

**Example 1**

```plaintext
DO SET=%%$S= THIS IS A SAMPLE STRING
DO SET=%%T= A1 A2 'SAMPLE' A3 A4 A5
DO SET=%%$PARSE %%S %%T
```

A match is found since the string SAMPLE is part of the original string.

The %%%$PARSRC System variable, which is discussed on page 247, can be used to check if all strings specified in the template were matched during the parsing process.
Using String Patterns

The original string is divided into two substrings while the matched part of the string is excluded. Parsing of the first substring will use the variables listed before the match on the template while parsing of the second substring will use the variables listed after the match:

— First substring: THIS IS A

As a result of parsing:

\[
\begin{align*}
A1 &= \text{THIS} \\
A2 &= \text{IS A}
\end{align*}
\]

— Second substring: STRING

As a result of parsing:

\[
\begin{align*}
A3 &= \text{STRING} \\
A4 &= \text{NULL} \\
A5 &= \text{NULL}
\end{align*}
\]

3. A match is not found. There is no substring identical to the given string pattern within the string.

It is assumed that a match is found at the end of the string. The first substring consists of the entire string and it is parsed using only the variables named before the string pattern on the template. Parsing continues from the character following the matched string (the end of the string, in this case).

**Example 2**

DO SET = %%S = THIS IS A SAMPLE STRING
DO SET = %%T = A1 A2 A3 'EASY' A4 A5
DO SET = %%$PARSE %%S %%T

A match was not found. The string ‘EASY’ does not exist within the original string.

— First substring: THIS IS A SAMPLE STRING

As a result of parsing:

\[
\begin{align*}
A1 &= \text{THIS} \\
A2 &= \text{IS} \\
A3 &= \text{A SAMPLE STRING}
\end{align*}
\]

— Second substring: NULL

As a result of parsing:
Using Numeric Patterns Within the Template

Numeric patterns are numbers that mark positions in the string. They are used to break the original string into substrings at the position indicated by the number.

The position specified can be absolute or relative:

- An absolute position is specified by an unsigned number.
- A relative position is specified by a signed number (positive or negative) and its purpose is to determine a new position within the string, relative to the last position.
- The last position is one of the following:
  - the start of the string (position 1), if the last position was not specified previously
  - the starting position of a string pattern if a match was found
  - the position of the end of the string, if the string pattern was not matched
  - the last position specified by a numeric pattern
- If the specified position exceeds the length of the string, the numeric pattern is adjusted to the end of the string. Similarly, if the specified position precedes the beginning of the string (negative or zero numeric pattern), then the beginning of the string is used as last position.

**Example 1**

A parsing template with an absolute numeric pattern:

```
DO SET=%%S =THIS IS A SAMPLE STRING
DO SET=%%T = A1 A2 11 A3 A4 A5
DO SET=%%$PARSE %%S %%T
```

- First substring: THIS IS A (up to, but not including, position 11)

As a result of parsing
Using Numeric Patterns Within the Template

A1=THIS
A2=IS A

- Second substring: SAMPLE STRING (from position 11, up to the end of the string).

As a result of parsing

A3=SAMPLE
A4=STRING
A5=NULL (0 length string)

Example 2

A parsing template with a relative numeric pattern:

```plaintext
DO SET=%%S=THIS IS A SAMPLE STRING
DO SET=%%T=A1 A2 +10 A3 A4 A5
DO SET=%%PARSE %%S %%T
```

Last position is the beginning of the string (position 1).

Position marked within the string is 1 + 10 = 11.

- First substring: THIS IS A (up to, but not including, position 11)

As a result of parsing

A1=THIS
A2=IS A

- Second substring: SAMPLE STRING (from position 11, up to the end of the string).

As a result of parsing:

A3=SAMPLE
A4=STRING
A5=NULL (0 length string)
Using More Than One Pattern and Combining Pattern Types in the Template

Both types of patterns (string and numeric) can be combined in the same template. Up to 30 patterns and up to 30 variable names can be specified.

Scanning of the string proceeds from beginning of the string until the first pattern (if any).

1. String pattern – A match was found

   The substring that precedes the match to the pattern is parsed using the variables named in the template before the pattern, with the last variable receiving the end of the substring, including leading and trailing blanks.

2. String pattern – A match was not found

   Since no match was found in the string, it is assumed that a match is found at the end of the string. The whole string is parsed using only the variables named in the template before the pattern.

3. Numeric pattern (absolute)

   The absolute numeric pattern points to a position within the string when the beginning of the string is position 1.

   The string is divided into two substrings.
   - The first substring extends from the beginning of the string and up to, but not including, the position that corresponds to the numeric pattern and it is parsed using the variables named in the template before the pattern.
   - If the absolute numeric pattern specifies a position beyond the length of the string, it is readjusted to the first position beyond the length of the string and the entire string is parsed using the variables named in the template before the pattern.

4. Relative numeric pattern

   The relative numeric pattern (a signed number) specifies a position within the string, relative to the last position.

5. Last position

   It is the beginning of the string when the relative numeric pattern is the first pattern in the template.
If the relative numeric pattern is not the first pattern in the template and the previous pattern was numeric, the last position is that specified by the previous numeric pattern.

If the relative numeric pattern is not the first pattern in the template and the previous pattern was a string, the last position is that of the starting character of the match (if there was a match) or the position following the end of the string (if there was no match).

As a result of what was just explained:

- If a pattern was not matched until the end of the string and the following pattern is a string pattern, this new string pattern is ignored since the starting point for the new scan is the end of the string.
- If a pattern was not matched until the end of the string and the following pattern is a numeric pattern, then the scan and subsequent parsing will resume from the new position indicated by that numeric pattern.

**Example 1**

A parsing template with two absolute numeric patterns (with the second position preceding the first):

The following DO SET statements:

```plaintext
DO SET=%%S = THIS IS A SAMPLE STRING
DO SET=%%T = A1 A2 11 A3 6 A4
DO SET=%%PARSE %%S %%T
```

have the same result as the following DO SET statements:

```plaintext
DO SET=%%A1 = THIS
DO SET=%%A2 = IS A
DO SET=%%A3 = SAMPLE STRING
DO SET=%%A4 = IS A SAMPLE STRING
```

- **First substring:** THIS IS A (up to, not including, position 11)
  
  As a result of parsing:
  
  A1=THIS
  A2=IS A

- **Second substring:** SAMPLE STRING (from position 11 and up to the end of the string; since the next pattern, position 6, precedes the previous position, it cannot limit this second substring)
As a result of parsing:
\[ A3=\text{SAMPLE STRING} \]

- Third substring: IS A SAMPLE STRING (from position 6 and to the end of the string)

As a result of parsing:
\[ A4=\text{IS A SAMPLE STRING} \]

**Example 2**

A parsing template with one absolute and one relative numeric pattern:

```plaintext
DO SET=%%S = THIS IS A SAMPLE STRING
DO SET=%%T = A1 6 A2 +3 A3
DO SET=%%PARSE %%S %%T
```

- First substring: THIS (beginning of the string up to, but not including, position 6).

As a result of parsing:
\[ A1=\text{THIS} \]

- Second substring: IS (from position 6 up to, but not including, position 6+3=9)

As a result of parsing
\[ A2=\text{IS} \]

- Third substring: A SAMPLE STRING (from position 9 to the end of the string)

As a result of parsing:
\[ A3=\text{A SAMPLE STRING} \]

**Example 3**

A parsing template with two relative numeric patterns:

The following DO SET statements

```plaintext
DO SET=%%T = A1 A2 +40 A3 -13 A4 A5
DO SET=%%S = THIS IS A SAMPLE STRING
DO SET=%%PARSE %%S %%T
```
have the same result as the following DO SET statements:

| DO SET=%%A1 = THIS  
| DO SET=%%A2 = IS A SAMPLE STRING  
| DO SET=%%A3 = %%NULL  
| DO SET=%%A4 = SAMPLE  
| DO SET=%%A5 = STRING |

The first numeric pattern specifies a position at column 40. This is beyond the end of the string so the position is reset to column 24 (end of the string + 1). As a result, the whole string is parsed to words using the A1 and A2 variables.

The second numeric pattern specifies a position at column 11 (end of the string + 1 minus 13) that precedes the position (40 readjusted to 24) previously specified; therefore the data from the last position (which is the end of the string) to the end of the string is parsed to words using the A3 variable (A3 is set to NULL).

The data (from column 12 to the end of the string) is parsed to words using the A4 and A5 variables.

**Example 4**

Combining a string pattern and numeric pattern

The following DO SET statements

| DO SET=%%S = THIS IS A SAMPLE STRING  
| DO SET=%%T = A1 'A' A2 +3 A3  
| DO SET=%%PARSE %%S %%T |

have the same result as the following DO SET statements:

| DO SET=%%A1 = THIS IS  
| DO SET=%%A2 = A S  
| DO SET=%%A3 = AMPLE STRING |

The pattern specifies a string (A) that is matched at column 9. The data before column 9 is parsed to words using the A1 variable. The Numeric pattern (+3) specifies a position at column 12 by using relative position. The data from column 9 to column 12 is parsed to words using the A2 variable. The remaining data (from column 12 to the end of the string) is parsed to words using the A3 variable.
Index

Symbols

$\text{SPACE}$ Member 71
% %PARSE Function 247

A

A Line Edit Command 238
Accessing a Table and its Rules
  Rule Definition Facility 65
ACTION Parameter 81
AND/OR/NOT Subparameter
  ON JOBEND Statement 57, 78, 112, 117, 121
  On Statements 80
AutoEdit Functions
  % %PARSE 247
Automated Console Action Parameter
  DO Statement 90

B

B Line Edit Command 238
B Option
  Table List Screen 70
BMC Software, contacting 2
Boolean Logic
  DO Statement 81
Boolean Operators
  DO IF Statement 93
BROWSE Option
  Table List Screen 65, 70

C

C Line Edit Command 237
CANCEL Command
  Report Decollating Mission Definition Entry Panel 67
CC Line Edit Command 237
CL Line Edit Command 237
COMMAND Parameter 57
Commands
  CANCEL 67
  Move 237
  Rule Definition Screen 82
  Rule Editing 235
CONSOLE ACTION Parameter
  DO Statement 81
CONTINUE SEARCH Parameter
  Description 88
Conventions Used in This Guide 14
Copy Commands
  Edit Environment 237
COPY Option
  Rule List Screen 65, 85
Copying a Rule to Another Table
  Rule Definition Facility 65
Copying Rules to Another Table
  Rule Definition Facility 85
Creating Rule Definitions
  Rule Definition Facility 64
CS Line Edit Command 237
CTJINIT
  Invoking from a batch JCL 176
  CTJINIT Parameters 175
  CTJINIT utility 173
CTJMRFLW 24, 129
  Invoking from a batch JCL 133
  CTJMRFLW Parameters 131
CTJPLAN
  Invoking from a batch JCL 138
CTJVER
  Invoking from a batch JCL 33
  CTJVER Parameters 30, 136
CTOPCMMSG Parameter 57
customer support 4

D

D (DELETE) Option
  Table List Screen 70
D Line Edit Command 236
DD Line Edit Command 236
DD Statement
  DASIM 30, 131, 136, 175
DELETE Command
  Edit Environment 236
DELETE Option
  Table List Screen 65, 70
Deleting
  Rule Definition Tables 70
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| Deleting a Table or Rule Table | Rule Definition Facility 65 |
| dependencies | 24, 129 |
| DESC Command | Rule Definition Facility 65 |
| DESC List Screen 73 |
| DESCRIPTION Parameter | Description 89 |
| DL Line Edit Command 236 |
| DO ELSE Statement 92 |
| Summary 81 |
| DO ENDF Statement 92 |
| Summary 81 |
| DO ENDM Strategy Statement | Summary 81 |
| DO IF Statement 92 |
| Boolean Operator 93 |
| Summary 81 |
| DO MSG 91 |
| DO MSG Parameter 90 |
| DO Option 81 |
| DO Parameter 90 |
| DO RETURN Statement | Summary 82 |
| DO REXX 82, 90, 97 |
| DO SET Statement | Summary 82 |
| DO Statement | Description 81, 90 |
| Parameter Summary 58 |
| DOMSG 62 |
| DORC 62 |
| DSN (Schedule library) statement keyword 132 |
| DSNEVENT Parameter 57 |
| Editing | Rule Definitions 82, 235 |
| ELSE Parameter 58 |
| ENDF Parameter 58 |
| Entry Panel | Rule Definition Facility 66 |
| EQ Operator | IF Statement 92 |
| EVENT Parameter 57 |
| Format of the Rule List Screen | Rule Definition Facility 72 |
| GE Operator | IF Statement 92 |
| GT Operator | IF Statement 92 |
| I | I Line Edit Command 237 |
| IEBCOPY 20, 31, 132, 138, 158, 170 |
| IEBGENER 20, 31, 132, 138, 158, 170 |
| IF Parameter 58 |
| IF Statement | EQ Operator 92 |
| GE Operator 92 |
| GT Operator 92 |
| LE Operator 93 |
| Logical Operators 92 |
| LT Operator 93 |
| NE Operator 92 |
| INIT 51 |
| INSERT Command | Edit Environment 237 |
| INSERT Option | Rule List Screen 64 |
| ISPF PACK Option | Rule Definition Facility 63 |
| J | JCL Library Mode | Parameters 30, 131, 136, 175 |
| JCL LIBRARY Parameter | AutoEdit Simulation 32 |
| JCL SYSIN | keywords 32 |
| job flow verification | 24, 129 |
| job order and dependencies | 24, 129 |
| job workflow and dependencies | 20 |
| JOBARRIV Parameter 57 |
| JOBEND Parameter 57 |
| L | LE Operator | IF Statement 93 |
| LIBRARY Parameter | CTMAESIM Utility 32, 132 |
| Line Editing Commands 236, 237 |
| Logical Operators | IF Statement 92 |
| LT Operator | IF Statement 93 |
| G | GE Operator | |

BMC Control-M Workload Automation JCL Verify User Guide
Index 261

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

M
M Line Edit Command 237
MEMBER Parameter
CTMAESIM Utility 32, 132
MESSAGE Parameter 57
Message/Event Selection Parameter
ON Statement 57
Message/Event Selection Parameters 79
ML Line Edit Command 237
MM Line Edit Command 237
Move Commands
Edit Environment 237
MS Line Edit Command 237

N
NE Operator
IF Statement 92
NEXTRULE Command
Rule Definition Screen 83
Numeric Patterns
Parsing 252

O
OMEGAEXP Parameter 57
ON Statement
Parameter Summary 57
Option OR
Primary Option Menu 64, 66
OPTLIST Command
Rule Definition Screen 82

P
Parameter
Rule Parameters 56, 78
Parameters
Description 59
Parsing 247
Combining Pattern Types 254
Numeric Patterns 252
Patterns 250
Words 248
PDSMAN 71
PREVRULE Command
Rule Definition Screen 83
product support 4

R
R Line Edit Command 237
Repeat Commands

Edit Environment 237
RESOLVE 62
RETURN Parameter 58
REXX EXECs 61
REXX Parameter 58, 90
RL Line Edit Command 237
RR Line Edit Command 237
RS Line Edit Command 237
Rule
Parameter Summary 57
Rule Definition
Creation 64
Deletion 65
Editing 235
Exiting 83
Maintaining Validity 238
Screen 63
Screen Commands 82
Screens 76
Rule Definition Facility
Accessing a Table and its Rules 65
Copying a Rule to Another Table 65
Copying Rules to Another Table 85
Deleting a Table of a Rule 65
Deleting Tables 70
Entry Panel 66
General 63
Rule List Screen 71
Saving Modifications 65
Search Window 67
Rule List Screen
Commands 73
COPY Option 65, 85
Exiting 84
Format 72
Options 73
Rule Definition Facility 71
RULE Parameter 57
Summary 57
Rule Parameters
Summary 76
Rule Types
Description 72
Runtime Scheduling Parameter
CONTINUE SEARCH 88

S
S Option
Table List Screen 70
Saving Modifications
Rule Definition Facility 65
SCHEDULING LIBRARY Parameter
AutoEdit Simulation 33
Screens
Rule Definition 63, 76
Rule Definition Entry Panel 66
Rule List 71
SELECT Option
Table List Screen 65, 70
SET Parameter 58
SETVAR 61
SMS Parameter 57
SMMSG Parameter 58
SORT 20, 31, 132, 138, 158, 170
STAT Command
Rule List Screen 73
STEP Parameter 57
String Manipulation
Parsing 247
STRING Parameter 57
support, customer 4
SUPUTIL 31, 132
SYSOUT Parameter 57

T
Table
Browse Mode 65
Deletion 65
Table List Exit Window
Table Creation 64
Table List Screen
DELETE Option 65
Options 70
Rule Definition 64
Rule Definition Facility 69
TABLE NAME Field
Rule Definition Entry Panel 65
TABLE Parameter
Testing AutoEdit Syntax 33
Tables
Deleting in Rule Definition Facility 70
technical support 4

U
UT 31, 132, 138, 170

V
Variables
Setting a Variable 98
Notes