

Control-0 Conversion Guide



Supporting

Version 7.0.00 of Control-0

September 2010



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

Address	BMC SOFTWARE INC 2101 CITYWEST BLVD HOUSTON TX 77042-2827 USA	Telephone	713 918 8800 or 800 841 2031	Fax	713 918 8000
----------------	------------------------------------------------------------------------	------------------	---------------------------------	------------	--------------

Outside United States and Canada

Telephone	(01) 713 918 8800	Fax	(01) 713 918 8000
------------------	-------------------	------------	-------------------

Copyright 2010 BMC Software, Inc., as an unpublished work. All rights reserved.

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.

DB2 is a registered trademark of International Business Machines Corporation.

The information included in this documentation is the proprietary and confidential information of BMC Software, Inc., its affiliates, or licensors. Your use of this information is subject to the terms and conditions of the applicable End User License agreement for the product and to the proprietary and restricted rights notices included in the product 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 UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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 Support page on the BMC Software website or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, please see “Before Contacting BMC Software.”

Support website

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

- read overviews about support services and programs that BMC Software offers
- find the most current information about BMC Software products
- search a database for problems similar to yours and possible solutions
- order or download product documentation
- report a problem or ask a question
- subscribe to receive e-mail notices when new product versions are released
- find worldwide BMC Software 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 support@bmc.com. Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC Software

Before you contact BMC Software, have the following information available so that Customer Support can begin working on your problem 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 problem
- 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.

Contents

About This Guide	9
How This Book is Organized	9
Conventions Used in This Guide	10
Chapter 1 Conceptual Overview	15
Overview	16
Automation Rule Definition	17
Rule Conversion	17
Rule Selection Criteria	18
Rule Description	18
Rule Scheduling	19
Rule Actions	19
Rule Owner ID	20
Rule Operation Mode	20
Rule Comments	20
Active and Inactive Rules	21
Automation Rule Interdependencies	21
Chapter 2 Conversion Steps	23
Overview	24
Step 1. Install the Conversion Utility	24
Step 2. Identify Key Components of Your Current Console Automation	25
Step 3. Gather Detailed Information About Your Current Console Automation.	25
Running the Utility	26
Step 4. Prepare a Work Plan	27
Location of Input for the Conversion Utility	27
Input from Different Products	27
Members not Containing Automation Definitions	28
Similar and Duplicate Definitions	28
Complexity of Automation Definitions	29
Step 5. Convert Automation Scripts using Utility CTOFANCI	29
Step 6. Review Output of CTOFANCI and Perform Manual Conversion Tasks	30
Step 7. Implement Converted Rules in a Test Environment	31
Step 8. Implement Converted Rules in a Production Environment	32
Chapter 3 Conversion Details	33
Conversion of Specific Parameters	34

Unique Control-O Parameters	36
MODE	36
OWNER.....	37
Chapter 4 Utility CTOFANCI	39
Overview	40
Logic	40
Input.....	40
Output	41
Monitoring Utility Execution	43
Restarting the Utility	43
Activating Utility CTOFANCI	43
Parameters.....	44
INCLUDE Statements	57
JCL Required to Execute CTOFANCI.....	58
Chapter 5 Reports	59
CTOFANCI Reports	60
RPTALL – General Message Report	60
RPTWARN – Warning Message Report	61
RPTFILES – Input/Output File Report	61
RPTRULES – Input Rule Report.....	62
RPTSRC – Input Rule Source Report.....	65
Title Lines	65
Text Lines	66
Sort-Stubs in Report RPTSRC.....	66
Appendix A Troubleshooting	69
Bypassing CTOFANCI Problems	69
Preventing Infinite Loops	69
Appendix B Problem Reporting	71
Reporting Infinite Loops	72
Appendix C Messages	73
Index	89

Tables

Symbols for Indicating Degree for Rule Conversion	18
Rule Operation Modes	20
Mechanisms for Defining Automation Rule Interdependencies	21
Conversion Steps	24
Reports Produced by Utility CTOFANCI	25
Tasks Performed by Utility CTOFNACI in Normal Mode	29
Conversion of CA-OPS Event Selection Criteria	34
Conversion of CA-OPS Time of Day Specification	34
Conversion of CA-OPS Action Statements	36
Conversion of CA-AUTOMATE Selection Criteria	36
Conversion of CA-AUTOMATE Keywords	36
Naming Convention and Location Parameters Created by Utility CTOFANCI	46
Report Allocation Parameters	50
Execution and Restart Parameters	51
Translation Parameters	55
CTOFANCI Reports	60
Fields in RPTFILES Report	61
Fields in the RPTRULES Report	63

About This Guide

This conversion guide reviews basic concepts relevant to converting from other console automation products to Control-O, and provides step-by-step instructions for converting to Control-O.



NOTE

This conversion guide is designed for use with Control-O version 5.0.0 and later.

Conversion to Control-O consists of executing a batch utility and performing a series of manual steps to adjust the utility's output. The batch utility runs independently of Control-O; however, to enable review and modification of conversion results using the INCONTROL™ Online facility, Control-O must be installed.

The conversion process:

- Produces detailed reports that help you plan and organize the conversion project.
- Builds Control-O Rule tables and rule definitions based on the rules and automation definitions of other console automation products.
- Enables you to automatically set unique Control-O parameters in the resulting Control-O rules.

How This Book is Organized

The guide is divided into the following chapters:

Chapter 1–Conceptual Overview

Overview of basic concepts relating to conversion to Control-O.

Chapter 2–Conversion Steps

A step-by-step procedure for performing a conversion from another product to Control-O.

Chapter 3–Conversion Definitions

Information about how parameters of other console automation products are converted to Control-O format.

Chapter 4–Utility CTOFANCI

Description of conversion utility CTOFANCI. This utility performs the automated steps of the Control-O conversion.

Chapter 5–Reports

Description of reports produced by utility CTOFANCI, and how they can be used to help manage the conversion process.

Appendix A–Troubleshooting

Information necessary for dealing with problems encountered during the conversion process.

Appendix B–Problem Reporting

Instructions on reporting problems to BMC Software Customer Support.

Appendix C–Messages

Messages produced by conversion utility CTOFANCI.

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

\$ is mapped to x'5B'

is mapped to x'7B'

@ is mapped to 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.

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

```
//PRDSYSIN DD * USERLOAD PRD(2) PRINT
```

- **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.

Conceptual Overview

This chapter includes the following topics:

Overview	16
Automation Rule Definition	17
Rule Conversion	17
Rule Selection Criteria	18
Rule Description	18
Rule Scheduling	19
Rule Actions	19
Rule Owner ID	20
Rule Operation Mode	20
Rule Comments	20
Active and Inactive Rules	21
Automation Rule Interdependencies	21

Overview

This conceptual overview describes how certain basic console management concepts are handled during conversion to Control-O. If you are not already familiar with Control-O concepts, it is highly recommended that you review the introduction chapter of the *Control-O User Guide*.

The Control-O Conversion Utility currently processes input from the following console automation products:

- CA-OPS
- CA-AUTOMATE
- MVS Message Processing facility (MPF)

The Conversion Utility also recognizes the following types of input, and copies the relevant members to specified output datasets and members.

- NetView AOC
- Documentation members
- MVS JCL
- Native TSO REXX
- Native TSO CLIST
- Certain high level languages (for example, C, PL/I)
- A program in ASM/370 or ASM/390

Future versions of the Control-O Conversion Utility will contain support for additional products and features as required.

NOTE



Space management members (for example, \$\$\$SPACE for PDSMAN) are ignored by the Control-O Conversion Utility.

The following pages describe how certain console automation issues are handled during conversion to Control-O. Special attention is given to the way in which information from current console automation definitions is converted to Control-O format.

Automation Rule Definition

The rule definition is the basic automation unit in Control-O. Each Control-O rule contains instructions that describe the action to be taken under specific circumstances.

The Control-O Conversion Utility accepts input consisting of various types of automation definitions and scripts of your current automation definitions. Batch utility CTOFANCI (provided with the Control-O Conversion Utility) uses this input to generate Control-O rules that perform similar functions. Control-O rules produced by the utility are stored in partitioned dataset (PDS) members called Rule tables. Control-O rules can be viewed and modified using the Control-O Rule Definition screen (Screen OR).

Rule Conversion

Conversion to Control-O consists of extracting automation information from your currently used product and translating it into Control-O rules that perform similar functions. The automatic step of the conversion process is performed by conversion utility CTOFANCI.

The basic component of automation information used as input for the conversion is the input rule. Each rule describes actions to be performed in a specific circumstance. One or more input rules may exist in a single input library. Input rules defined for multiple automation products can be processed in a single run of the conversion utility.

The main output of the conversion utility consists of Control-O rules and Rule tables. The Control-O rules produced as part of the conversion process are referred to as “output rules” throughout this guide.

Below is a description of the basic components of Control-O rules and how they are automatically generated from information in the input libraries by conversion utility CTOFANCI.

Rule Selection Criteria

Selection criteria of a rule determine the event (or events) that must trigger the actions specified in the rule. In Control-O these criteria are specified in ON statements. ON statements in output rules created by utility CTOFANCI are derived from event definitions in members specified as input for utility CTOFANCI. Input rules (that is, rules from other products) that are activated at a specific time are converted to Control-O EVENT rules (that is, triggered using an ON EVENT statement) with appropriate scheduling and TIME parameters.

Rule Description

Each Control-O rule contains a DESCRIPTION field of one or more lines. The DESCRIPTION field of rules created by utility CTOFANCI contains the following elements:

- Indicator of the degree to which the rule was converted. The following symbols may be displayed:

Table 1 Symbols for Indicating Degree for Rule Conversion

Symbol	Description
<<<<	This output rule is a complete translation of an input member. The rule must be reviewed to ensure that the result of the automatic rule creation accomplishes the desired action.
<<<-	A substantial amount of translation was performed. However, certain lines of the input script contained elements that were not converted by utility CTOFANCI. These elements must be converted manually.
<<--	Less than half the input member was translated by utility CTOFANCI.
<---	Very basic transformation was performed (for example, only ON statements were automatically generated).

After you manually adjust a rule, you can replace the indicator described above with an indicator that reflects the rule's new status. The updated indicators can help track the progress of your conversion effort.

- Identification of the input rule from which the output rule was created. The following information is supplied.
 - Name of the input dataset.
 - Name of the input member.
 - If more than one rule was generated from an input member, a sequence number is specified for the rule.
- Intended output name (if a duplicate name was generated). For more information about duplicate names, see [“Output Member Names” on page 42](#).
- Date and time the Control-O rule was created.
- Line with blank fields to be filled in by the reviewer (for example, to indicate the date and time the rule was reviewed).

Rule Scheduling

Powerful scheduling capabilities are a key feature of INCONTROL and Control-O. Most other console automation products provide time-driven rules. However, they do not allow specification of generalized schedules (calendars) that determine which automation rules must be applied to which dates, days, and calendar functions.

The Control-O Conversion Utility translates time dependent functions to Control-O scheduling parameters. In the manual adjustment phase of the conversion, you can enhance automation definitions using additional scheduling parameters and user-defined calendars, described in the online facilities chapter in the *Control-O User Guide*.

Rule Actions

Actions performed by Control-O rules are specified using DO statements. Future versions of the Control-O Conversion Utility will recognize action statements of other console automation products and translate them into DO statements in the output rules. In the current version, only message suppression is recognized and automatically translated by the Conversion Utility.

For more information about the various DO statements, see the rule parameters chapter in the *Control-O User Guide*.

Rule Owner ID

An owner ID is associated with each Control-O rule. This ID is specified in parameter OWNER of each Control-O rule definition. Parameter OWNER in Control-O rules is set to the value specified for parameter RULEOWNER of utility CTOFANCI.

To specify different owner IDs for rules created by utility CTOFANCI, run the utility more than once. For each run of the utility, specify different input members and different values for parameter RULEOWNER.

Rule Operation Mode

An operation mode is associated with each Control-O rule. This mode is specified in parameter MODE of each Control-O rule definition. The following modes can be specified:

Table 2 Rule Operation Modes

Mode	Description
PROD	Normal operation mode.
LOG	Same as normal operation mode, but descriptive messages are written to a log file.
TEST	Test mode. Operations are not performed, but messages are written to a test log describing what Control-O would have done if the rule was in PROD mode.

Parameter MODE in Control-O rules is set according to the value specified for parameter RULEMODE of utility CTOFANCI.

Rule Comments

Statements of the input member used to create a Control-O rule are included as comments in the output Control-O rule. Optionally, the comment lines in the input member can also be copied to the output rule.

Utility CTOFANCI handles input members stored in a variety of file formats (for example, FB or VB) and record lengths. While inserting lines from an input member as comments in a Control-O rule, the CTOFANCI utility may adjust the lines to fit in the rule. This adjustment may split an input line into one or more comment lines, or truncate strings of identical characters (for example, blank, '*' or '-').

Active and Inactive Rules

Control-O rules created during the conversion process can be organized in a number of different Rule tables (PDS members). Control-O determines which Rule tables must be activated by referencing Rule Table List member RULELIST when Control-O is initialized.

If you do not want to activate the rules contained in a specific table or tables, ensure that these tables are not listed in member RULELIST.

For more information about the Rule Table list, see the *INCONTROL for z/OS Administrator Guide*.

Automation Rule Interdependencies

Although many automation rules are independent of other rules, the implementation of some rules depend on actions performed using other automation rules. In Control-O, these interdependencies can be defined using the following mechanisms.

Table 3 Mechanisms for Defining Automation Rule Interdependencies

Mechanism	Description
Prerequisite conditions	A condition or conditions set by a rule to enable execution of another rule.
DO RULE statement	When specified in a Control-O rule, this statement can trigger another rule (in which a corresponding ON RULE statement is specified).
DO TSO statements	When specified in a Control-O rule, this statement can invoke a TSO, REXX procedure or CLIST program.

Conversion Steps

This chapter includes the following topics:

Overview	24
Step 1. Install the Conversion Utility	24
Step 2. Identify Key Components of Your Current Console Automation	25
Step 3. Gather Detailed Information About Your Current Console Automation. . . .	25
Running the Utility	26
Step 4. Prepare a Work Plan	27
Location of Input for the Conversion Utility	27
Input from Different Products	27
Members not Containing Automation Definitions	28
Similar and Duplicate Definitions	28
Complexity of Automation Definitions	29
Step 5. Convert Automation Scripts using Utility CTOFANCI	29
Step 6. Review Output of CTOFANCI and Perform Manual Conversion Tasks	30
Step 7. Implement Converted Rules in a Test Environment	31
Step 8. Implement Converted Rules in a Production Environment	32

Overview

This chapter describes the steps of the conversion process. A q marker indicates the beginning of each step. Read through this chapter before performing the steps. It is important to follow the sequence of the steps described below, to ensure a successful conversion.

The following summary checklist is provided for your convenience:

Table 4 Conversion Steps

Step	Description
1.	Install the Conversion Utility
2.	Identify Key Components of Your Current Console Automation
3.	Gather Detailed Information About Your Current Console Automation
4.	Prepare a Work Plan
5.	Convert Automation Scripts using Utility CTOFANCI
6.	Review Output of CTOFANCI and Perform Manual Conversion Tasks
7.	Implement Converted Rules in a Test Environment
8.	Implement Converted Rules in a Production Environment

Step 1. Install the Conversion Utility

The Conversion Utility is supplied as a binary file on a diskette, using FTP, or as an e-mail attachment. Perform the following steps to install the Control-O Conversion Utility at your site:

- 1** Allocate a sequential file on the mainframe computer. Specify LRECL=80 and BLKSIZE=22000.
- 2** Transfer the binary file from the PC to the sequential file created in the previous step. Ignore carriage-return and line-feed (CR/LF) characters and do not perform ASCII-to-EBCDIC conversion during the file transfer.
- 3** Convert the sequential file to a partitioned dataset (PDS) using the following RECEIVE command:

```
TSO RECEIVE INDATASET(' dsname')
```

where *dsname* is the name of the sequential file.

You are prompted for additional parameters of the RECEIVER command. Respond as follows:


```
DATASET( ' pdsname ' )
```

where *pdsname* is the name of the PDS in which the conversion programs should be stored. (This PDS is created by the RECEIVE command.)

Step 2. Identify Key Components of Your Current Console Automation

Gather the following information about console automation at your site:

- Names and/or prefixes of libraries containing automation scripts.
- Initialization parameters specified for your current console automation definitions.
- Console automation requirements at your site (that is, what functions are currently handled by console automation tools, and what additional functions, if any, you want Control-O to handle).

Step 3. Gather Detailed Information About Your Current Console Automation

Run utility CTOFANCI in simulation mode to generate information about your current console automation definitions.

Specify the libraries containing your current console automation definitions as input for the utility.

The utility produces reports that provide detailed information about your console automation definitions and how they are converted to Control-O format.

The following reports are provided:

Table 5 Reports Produced by Utility CTOFANCI (part 1 of 2)

Report	Description
RPTALL	General Message Report. Contains a copy of the control statements of the utility, and all messages generated by the utility.
RPTWARN	Warning Message Report. Contains all warning messages issued by the utility.
RPTFILES	Input/Output File Report. Contains information about all input and output files (datasets) used by the utility.

Table 5 Reports Produced by Utility CTOFANCI (part 2 of 2)

Report	Description
RPTRULES	Rule Report. Contains information about all input rules processed by the utility (for example, event type, rule size, and output member name).
RPTSRC	Rule Source Report. Contains all automation definition statements of input rules that were processed by the utility.

For a detailed description of utility CTOFANCI, see [Chapter 4, “Utility CTOFANCI.”](#)

For more information about reports produced by utility CTOFANCI, see [Chapter 5, “Reports.”](#)

Running the Utility

Perform the following steps to run utility CTOFANCI at your site.

1 Prepare the control statements of the utility.

- A** Edit member SYSINPRM (supplied with the Conversion Utility) or a copy of this member. This member contains the control statements for utility CTOFANCI. (For valid control statements and parameters that can be specified for the utility, see [Chapter 4, “Utility CTOFANCI”](#)).
- B** Save the member.

2 Prepare the conversion job.

- A** Edit member CTOFJOB (supplied with the Conversion Utility). This member contains a job that invokes REXX program CTOFANCI in TSO batch mode.
- B** Modify the SYSEXEC and SYSPROC DD statements to reference the library containing the CTOFANCI REXX execs.
- C** Modify the last line of member CTOFJOB so that it references a dataset and member name containing the control statements of utility CTOFANCI.
- D** Save the member CTOFJOB.

3 Submit job CTOFJOB.

For information about monitoring the execution of the utility see [Chapter A, “Troubleshooting.”](#)

Multiple runs of utility CTOFANCI can be used to produce reports about different groups of input libraries and members and to adjust parameters to determine how they will affect conversion at your site.

Step 4. Prepare a Work Plan

Use the information collected in Steps 2 and 3 (above) to determine what will need to be done as part of the conversion process.

Consider the following issues when developing a work plan for conversion of console automation at your site to Control-O format:

- Location of Input for the Conversion Utility
- Input from Different Products
- Members not Containing Automation Definitions
- Similar and Duplicate Input Scripts
- Complexity of Automation Definitions

Each of these issues is discussed in detail below.

Location of Input for the Conversion Utility

Your current console automation definitions may be located in any number of libraries. Some libraries may have some members that contain console automation definitions, and other members that do not contain such definitions.

The way in which automation definitions are organized at your site can influence the method you choose for converting them. Conversion utility CTOFANCI can be used to convert automation definitions in one or more libraries in a single run. (for more information, see the following section.)

Input from Different Products

If several console automation products are in use at your site, you can convert the members containing automation definitions for multiple products at the same time, or for one product at a time.

- If each product was used to implement automation of a different aspect of your environment, it is recommended that you convert the rules for each product separately.

- If a certain type of automation has been implemented using more than one product at your site, it is recommended that you process all relevant libraries at the same time. This method can result in a more compact and more easily managed set of output rules.

The RPTRULES and RPTSRC reports generated by utility CTOFANCI can be sorted so that output rules are grouped by function (for example, all rules triggered by a specific message). This facilitates combination of automation rules from different products into Control-O rules that perform all the required functions.

Members not Containing Automation Definitions

Your automation libraries may contain members that do not contain automation scripts. These members may contain programs, CLISTs, REXX scripts, or JCL jobs that are part of your automation applications, or they may be related to other activities in your site.

You can identify these members and their locations using reports RPTRULES and RPTSRC, which are generated during a simulation run of the conversion utility CTOFANCI (performed in Step 3 above).

When utility CTOFANCI is run in normal mode, you can indicate names or a name pattern for output datasets and members to which members not containing automation definitions should be copied.

Similar and Duplicate Definitions

You can have more than one automation definition at your site that performs a given task. You can also have multiple definitions for similar tasks. Duplicate or similar definitions may have been defined for different systems, or they may be copies of a rule that are no longer in use. Sometimes, nearly identical rules perform the same function on different messages.

You should consolidate automatic conversion of similar automation definitions. During the manual adjustment phase of the conversion process you can delete unneeded duplicate Control-O rules that were created during the automatic phase of the conversion. Combining the conversion of similar automation definitions can result in more efficient Control-O implementation.

Report RPTRULES can be sorted according to various fields. The sorted report can be used to identify automation definitions that perform the same (or similar) tasks. The following attributes can be used to identify duplicate or similar input members:

- Similar member names in different libraries.

- Identical member size (total number of lines or number of member lines excluding comments).
- Similar selection criteria (for example, rules that are triggered by the same message ID, or at the same time of day).

Complexity of Automation Definitions

Automation definitions can be very simple (for example, unconditional suppression of a specific message) or they can involve complex logic.

One way to identify complex automation definitions is by sorting the input scripts (rules) at your site according to length. Input script length is listed in reports RPTRULES and RPTSRC (for details, see [“Step 3. Gather Detailed Information About Your Current Console Automation”](#) on page 25).

Step 5. Convert Automation Scripts using Utility CTOFANCI

Run utility CTOFANCI in normal mode.

Utility CTOFANCI performs the automatic phase of conversion to Control-O. When run in normal mode, this utility processes your inventory of console automation definitions and performs the following tasks:

Table 6 Tasks Performed by Utility CTOFANCI in Normal Mode

Task	Description
Library and table creation	Utility CTOFANCI creates output libraries and Control-O Rule tables to contain output members and rules produced by the utility. Parameters of the utility specify the composition of the names of output libraries and Rule tables.
Rule creation	Utility CTOFANCI creates Control-O rule definitions that can be used to implement automation tasks similar to your current console automation definitions.
Inventory analysis	A series of reports are produced by each run of utility CTOFANCI. These reports can be used to analyze your current automation definitions. Reports produced by utility CTOFANCI can also be used to determine how utility parameters should be modified for additional runs of the utility. CTOFANCI reports are described in detail in Chapter 5, “Reports.”

Multiple runs of utility CTOFANCI can be used to process different groups of input libraries and members and to adjust parameters to suit your conversion needs.

For information about how to run utility CTOFANCI, see “[Step 3. Gather Detailed Information About Your Current Console Automation](#)” on page 25.

Step 6. Review Output of CTOFANCI and Perform Manual Conversion Tasks

The output of utility CTOFANCI (run in Step 4 above) consists of:

- Control-O rules created from information in the input libraries.
- Members that do not contain automation definitions and were copied from the input libraries to new locations.
- Reports describing input and output of the utility. Certain reports are mentioned briefly below. These reports are described in detail [Chapter 5, “Reports.”](#)

Perform the following steps to review the output of utility CTOFANCI:

1. Review the list of output rule libraries created or updated by utility CTOFANCI. This list is provided in report RPTFILES.
2. Examine the Control-O rules created by utility CTOFANCI. (These rules are located in the rule libraries identified in the RPTFILES report.) The input script used to create each Control-O rule is included as comments in the rule. Review the input script (in the comments) and verify that statements converted automatically by the utility conform to your site’s requirements.

Review statements of the input scripts not converted by the utility (in the comment lines of the output rules), and specify Control-O rule definition parameters that provide the required functionality.

To help identify rules that must be modified, you can sort report RPTRULES by output dataset name, member name, and rule number to prepare an output rule verification checklist.

You can extract report lines that contain a warning indicator to identify rules that require extra attention. You can assign rule verification tasks according to the completion indicator (<--, <<--, <<<-, or <<<<, where rules with an <<<< indication need only be visually verified while other rules may require more manual work).

For more information about Control-O rule definition parameters, see the rule parameters chapter in the *Control-O User Guide*.

NOTE



If error message IOAE33E (Insufficient Storage) is displayed during an attempt to view a Rule table online, you can correct this problem by splitting the Rule table into multiple tables.

3. Review report RPTWARN and make sure that each warning in this report has been handled. Possible responses to warnings in this report are:
 - No response, if the default handling performed by the utility satisfies your site's functional requirements.
 - Modify the output rules (for example, using specification of additional rule definition parameters in the rules) to implement functionality described by input script statements, keywords, or parameters that were not translated by the utility.
4. Rename output members that have duplicate names. Utility CTOFANCI assigns these members a prefix of @DUP (or an alternative prefix specified using utility parameter DUPREF). The new names assigned to these members should conform to your site's naming conventions.

NOTE



Certain duplicate members may not be needed. Unnecessary duplicate members can be deleted at this stage, or, if necessary, combined with other members to preserve desired functionality. (For more information, see [“Similar and Duplicate Definitions”](#) on page 28.)

Step 7. Implement Converted Rules in a Test Environment

Use the following steps to test the Control-O rules that you created in Steps 4 and 5:

1. Install INCONTROL and Control-O in a test environment, where you can perform operations without adversely affecting production at your site.
2. Specify the rule libraries created during conversion in the Rule Table list (member RULELIST) referenced by DD statement DARULLST in the Control-O monitor procedure.
3. Activate Control-O.

4. Generate events (for example, messages or commands) that should trigger Control-O rules. For example:
 - Generate messages and commands using utility IOATEST or CTOTEST.
 - Trigger time-driven rules by modifying the time of day in the system (this may require an IPL).
 - Submit jobs to trigger rules with selection criteria specifying job-related events (for example, job arrival, job ending, or job step completion).
5. Verify that the Control-O rules handled simulated events in the desired way.
 - Watch the console for messages that should be issued or suppressed by the rules.
 - Watch TSO user IDs that should receive notification of events (that is, Shout messages).
 - Review the JES console log and the Control-O Automation log for additional messages that describe how events in the test environment were handled.

Step 8. Implement Converted Rules in a Production Environment

Use the following steps to implement Control-O rules created by the conversion process:

1. Install INCONTROL and Control-O in your production environment.
2. Specify the rule libraries created during conversion in the Rule Table list (member RULELIST) referenced by DD statement DARULLST in the Control-O monitor procedure.
3. Activate Control-O.

NOTE



The user can run specific Control-O rules in TEST mode to test the performance of the converted rules in the production environment.

Conversion Details

This chapter includes the following topics:

Conversion of Specific Parameters	34
Unique Control-O Parameters	36
MODE	36
OWNER	37

Conversion of Specific Parameters

This chapter contains details about the conversion of specific keywords and parameters of console automation products at your site into Control-O rule definition parameters.

This chapter also describes unique Control-O parameters that can be specified using parameters of conversion utility CTOFANCI.

For an in-depth description of all Control-O Rule Definition parameters, see the Rule Parameters chapter in the *Control-O User Guide*. The conversion tables below list keywords and parameters of other console automation products and the Control-O parameters to which these keywords and parameters are converted.

Table 7 Conversion of CA-OPS Event Selection Criteria

CA-OPS Event Selection Criteria	Control-O Rule Definition Parameter
)CMD	ON COMMAND
)DOM	ON EVENT
)EOM	ON JOBEND
)GLV	ON EVENT
)MSG	ON MESSAGE
)OMG	ON EVENT
)REQ	ON EVENT
)SCR	ON EVENT
)SEC	ON EVENT
)TOD	ON EVENT, WDAYS, TIME, INTERVAL For details, see Table 8 on page 34 .

Table 8 Conversion of CA-OPS Time of Day Specification

CA-OPS Time of Day Keywords	Control-O Rule Definition Parameter	Note (see after table)
)TOD <i>fromtime totime</i>	TIME FROM / UNTIL	1
)TOD <i>interval</i>	INTERVAL	2
)TOD <i>fromweekday toweekday</i>	WDAYS	1
)TOD <i>fromdate</i>	Not yet supported	
)TOD <i>todate</i>	Not yet supported	

Note 1

Control-O TIME FROM and UNTIL fields can only be used to specify limits within a single day (24 hours).

If a combination of times and weekdays was specified in an input rule and the time limits are within a single day, these time limits are translated correctly.

If a time limit that exceeds a single day is specified in the input rule, a warning message is issued. The time limits specified in the input rule are translated as if they were specified separately for each day.

For example, the following CA-OPS specification:

```
MONDAY 05:00, ,TUESDAY 10:00
```

is translated to Control-O as:

```
WDAYS 1,2
TIME FROM 0500 UNTIL 1000
```

Note 2

The Conversion Utility translates rule activation intervals as follows:

- If the interval in the input rule is smaller than 999 minutes, the same interval is entered in the INTERVAL field in the output rule definition.
- If the interval in the input rule is more than 999 but less than 1440 minutes (one day), a warning message is issued and an interval of 999 minutes is specified in the output rule.
- If the interval in the input rule is exactly one day or exactly one week, the interval is ignored and a warning message is issued. Because Control-O rules are typically reloaded at the beginning of each working day, the output rule can implement the same interval without a value for parameter INTERVAL being specified in the output rule.
- If the interval in the input rule is more than one day, but not exactly one week, a warning message is issued and this rule must be modified manually to provide the desired functionality.
- If the interval specified for the input rule includes partial minutes (for example, 150 seconds), the output interval is equal to the number of whole minutes in the input interval. For example, 150 seconds is translated to 2 minutes.

If the number of whole minutes is zero (for example, 30 seconds was specified), an interval of one minute is specified in the output rule.

Table 9 Conversion of CA-OPS Action Statements

CA-OPS Statement	Control-O Rule Definition Parameters
)PROC RETURN "SUPPRESS"	DO DISPLAY SUPPRESS Y
)MSG <i>msgid</i> NOOPSLOG	DO SET %%%\$AUTOLOG=NO

Table 10 Conversion of CA-AUTOMATE Selection Criteria

CA-AUTOMATE	Control-O Event Selection Criteria
MSGID()	ON MESSAGE
CMDIN()	ON COMMAND
SCREEN()	ON EVENT
STATVAR()	ON EVENT
TIME()	ON EVENT

Table 11 Conversion of CA-AUTOMATE Keywords

CA-AUTOMATE Keyword	Control-O Event Selection Criteria
SUPPRESS	DO DISPLAY SUPPRESS Y
SUPPRESS(NOLOG)	DO DISPLAY SUPPRESS A
TIME(HH:MM)	TIME FROM HHMM TO 2359
EVERY(<i>interval</i>)	INTERVAL (For more information, see "Note 2" on page 35) APPEARED n TIMES IN m MINUTES

Unique Control-O Parameters

Certain unique Control-O rule definition parameters can be set by the Conversion Utility in the process of creating the Control-O Rule tables. Values for these parameters can be specified using a parameter of conversion utility CTOFANCI, or manually following automatic rule creation.

The following unique Control-O parameters are available:

MODE

This parameter indicates the mode in which a Control-O rule must operate. Valid values:

- PROD – Normal operation mode.
- LOG – Same as PROD, except that in addition to normal functions, all actions are written to a test journal.
- TEST – Console actions are not performed, but messages are written to a test journal, describing actions that would have been performed if the rule was in PROD mode. Setting of variables is performed.

The value for parameter mode in each output rule is determined by parameter RULEMODE of utility CTOFANCI. For more information, see [Chapter 4, “Utility CTOFANCI.”](#)

OWNER

This parameter specifies a user ID or user name to be associated with each Control-O rule. The value specified for this parameter can be used by Control-O security to check authorization to perform the actions specified in the rule.

The OWNER value for rules generated during conversion is determined by parameter RULEOWNER of utility CTOFANCI. For more information, see [Chapter 4, “Utility CTOFANCI.”](#)

Utility CTOFANCI

This chapter includes the following topics:

Overview	40
Logic	40
Input	40
Output	41
Monitoring Utility Execution	43
Restarting the Utility	43
Activating Utility CTOFANCI	43
Parameters	44
INCLUDE Statements	57
JCL Required to Execute CTOFANCI	58

Overview

This chapter contains a detailed description of utility CTOFANCI. This utility performs many of the primary tasks necessary for conversion to Control-O.

Using utility CTOFANCI, you can:

- Generate reports that you can use to develop a conversion work plan.
- Create output libraries and copy or convert input members to members in the appropriate output libraries.
- Translate specific automation keywords and parameters in input rules to Control-O rules with similar functionality.

Additional parameters of this utility enable you to:

- Specify naming conventions and space management for libraries and members created by the utility.
- Indicate whether to copy comment lines in input members to the output members.
- Divide large amounts of input into multiple smaller runs of the utility and/or consolidate results from multiple runs of the utility.
- Run the utility in simulation mode to gather information needed for a conversion work plan.

Logic

Input

Input to utility CTOFANCI consists of the libraries (PDSs) at your site that contain your currently implemented console automation definitions.

Input datasets and, optionally, members are specified to utility CTOFANCI using INCLUDE statements (see [“INCLUDE Statements” on page 57](#)).

Input datasets are handled one at a time. All selected members in each input dataset are processed before the utility proceeds to the next dataset.

Output

The content of each input member is analyzed to determine the language or automation product with which it is associated (for example, CA-AUTOMATE, CA-OPS, REXX program, or documentation member).

- Input members containing console automation definitions are used to create Control-O rules with similar functionality. Each rule is added to a specified output Rule table (PDS member). Depending on utility parameters, more than one Rule table may be created. Additionally, more than one rule with the same ON statement (that is, selection criteria) may be created.

For information on conversion of specific keywords and parameters in the input scripts, see [Chapter 3, “Conversion Details.”](#)

- Input members containing other information (for example, CLISTs, TSO REXX or documentation members) are copied to locations specified using parameters of the utility. Each of these members is copied to a separate member in an output library. A comment is added to the beginning of each member containing the input member name and the date and time the member was processed by utility CTOFANCI. For example:

```
/* <--- SITE1.REXX(REXX1) 00/10/13 09:02 */
```

The name of each output dataset and member is determined by the name pattern specified for parameters OUTDSN and OUTMEMNAME (described in [“Output Parameters” on page 45](#)).



NOTE

Certain members in input libraries are neither translated nor copied into output libraries. Examples of such members are PDSMAN control members (for example \$\$\$\$SPACE) and Control-O rules (if any) in the input libraries.

Output Datasets

Output dataset names are determined according to a pattern specified using parameter OUTDSN (described in [Table 13 on page 46](#)).

- If the output dataset does not exist, the utility creates a new PDS with the specified name and allocation parameters.

- If the resolved dataset name matches the name of an existing dataset, processing depends on the value specified for utility parameter RESET.
 - By default, the existing dataset is used.
 - If RESET=OUTLIB is specified, a new dataset is created with the same name appended by a unique suffix. (The suffix appended to the new dataset name is the same as the <RUNID> keyword described in [Table 13 on page 47.](#))

Output Member Names

Each output member created by utility CTOFANCI is assigned a name generated by resolving a member name pattern specified using parameter OUTMEMNAME (described in [Table 13 on page 48.](#))

- If the intended name of an output member matches the name of a member already in the designated output library, the new member is handled in the following way:
- If a rule is directed to an existing output member, and that member is a Rule table, the rule is added to the Rule table.
- If the new output member is not a rule member (for example, it is a TSO REXX or a Documentation member), the output is placed in a new member with a name of:

@DUPnnnn

where:

- *@DUP* is a duplicate member prefix specified using parameter DUPPREFIX (Default: @DUP).
- *nnnn* is a sequential number from 0001 to 9999.
- If a rule is directed to an existing output member, but that member is not a Rule table, utility CTOFANCI searches among the existing @DUPnnnn members for a Rule table whose intended name matches the desired location for the rule. If such a table is found, the rule is added to the table. If no such table is found, a new @DUPnnnn member is created (using the preceding name format).

Reports

In addition to the rule libraries created by utility CTOFANCI, a variety of reports are produced by each run of the utility. These reports provide information about datasets and members selected by the utility, and how they were processed by the utility.

Reports generated by a run of the CTOFANCI utility (in normal or simulation mode) can be sorted to organize information that helps you develop a work plan for conversion of your site's console automation definitions to Control-O.

For more information about reports produced by utility CTOFANCI, see [Chapter 5, “Reports.”](#)

Monitoring Utility Execution

You should monitor the execution of utility CTOFANCI (by browsing the utility output) so that you can handle infinite loops promptly if they occur.

For information about how to handle infinite loops in a run of utility CTOFANCI, see [Chapter A, “Troubleshooting.”](#)

Restarting the Utility

If a previous run of utility CTOFANCI ended early (due to errors or because a specified maximum number of input members were processed), you can avoid unnecessary repetition of processing already performed by indicating that the utility begin processing with a specific dataset or member. For example, you can indicate that processing continue with the last dataset processed by the utility, or with the dataset or member that should have been processed next. For more information, see [“Execution and Restart Parameters” on page 51.](#)



NOTE

For performance reasons, utility CTOFANCI often accumulates several rules—which are destined for the same Rule table—in memory before writing them to disk. If utility CTOFANCI ends due to an error, rules still in memory may appear in the utility’s reports, but may not have been completely processed.

If CTOFANCI ends abnormally, it is recommended that you check the last rules processed by the utility to determine if incompletely processed rules listed in the utility reports are missing from the Rule tables. Ensure that the restart of the utility reprocesses any rules that were not completely processed in the previous run of the utility.

Activating Utility CTOFANCI

For more information about how to invoke utility CTOFANCI, see [Chapter 2, “Conversion Steps.”](#)

The following control statements show all parameters that can be specified for utility CTOFANCI. This section describes each parameter as described in detail.

Figure 1 Control Statements for Utility CTOFANCI

```

TYPERUN  OUTDSN=outdsn,OUTMEMNAME=memname,MODE={NORMAL|SIMULATION},
         [OUTVOL=volser,][OUTUNIT=unitname,]
         [OUTTRKS1=tracks,][OUTTRKS2=tracks,]
         [OUTDIR=dirblocks,][RLSE=Y|N,]
         [SYSOUT=sysout-class,][RPTSRCDSN=dsn|DUMMY,]
         [RPTRULESDSN=dsn,][RPTRULESMOD=Y|N,]
         [RPTSRCMOD=Y|N,][RESET=OUTLIB|NONE,]
         [OUTBLKSZ=block-size,][OUTCOMMENTS=Y|N,]
         [RESTARTDS=dsname|dsnid,] [SWEET=SUN|MON,]
         [DAYTIME=+HHMM|-HHMM,]
         [RESTARTMEM=memname,][RULEPROGRESS=nn,]
         [LINESEQ=NONE|STD|AUTO|col1-col2,][SIMLINELIM=nnn,]
         [RULEWARNLIM=nn,][STOPAFTER=nnn,]
         [DUPPPREF=prefix,][RULEMODE=T|P,][RULEOWNER=ownerid]
INCLUDE  DSN=dsname,MEMBER=memname
INCLUDE  ...

```

The following rules apply to the syntax for control statements specified for utility CTOFANCI:

- Multiple INCLUDE statements can be specified for the utility.
- Each control statement (that is, a TYPERUN or INCLUDE statement) can span multiple lines.
- Specification for each parameter in a control statement must be contained in a single line.
- If not all parameters of a statement fit on a given line, a comma must appear after the last parameter on the line and subsequent parameters must be specified on subsequent lines. Positions between the comma and the end of the line must contain only blank spaces.
- To enter a comment line, specify an asterisk (*) in column 1 of the line.

Parameters

The parameters of utility CTOFANCI described in this section are specified using the TYPERUN statement. Only one TYPERUN statement can be specified for each run of this utility.

The TYPERUN statement can be used to specify:

- Output Parameters
- Report Allocation Parameters
- Execution and Restart Parameters
- Translation Parameters

Output Parameters

The following parameters indicate the naming conventions and locations for datasets and members created by utility CTOFANCI.

Table 12 Naming Convention and Location Parameters Created by Utility CTOFANCI (part 1 of 4)

Parameter	Description
OUTDSN	<p>Specifies values that determine the name of the output library for each rule created by the utility. Mandatory.</p> <p>The value specified for this parameter can be a maximum of 72 characters. The value can include any combination of constants (that is, A-Z, 0-9, \$, @, or '.') and/or one or more of the following keywords, including the angle brackets (<>):</p> <ul style="list-style-type: none"> ■ <INSDN>—Name of the dataset containing the input rule. ■ <INMEM>—Name of the input member. ■ <INQn>— The nth qualifier of the input dataset name (for example, if the input rule is in a PDS named A.BB.CCC, then <INQ3>=CCC). ■ <INQLn>—The nth last qualifier in the input dataset name (for example, if the input is in a PDS named A.BB.CCC, then <INQL3>=A). ■ <PRODCODE>—Product code of the input member. Each product or script language is associated with a 3-character code. The following codes are used: <ul style="list-style-type: none"> — AOC—NetView AOC — ASM—A program in Assembler 370 or 390 — AUT—CA-AUTOMATE — CLS—Native TSO CLIST — DOC—Documentation — JCL—MVS JCL — HLL—A high level language (for example, C or PL/I) — MPF—MVS Message Processing facility — OPS—CA-OPS — REX—Native TSO REXX — UNK—Unknown product

Table 12 Naming Convention and Location Parameters Created by Utility CTOFANCI (part 2 of 4)

Parameter	Description
OUTDSN (continued)	<ul style="list-style-type: none"> ■ <ETYPE>--The event type of the input member or rule. The utility automatically assigns one of the supported event types to each input member or rule. The following event types are used: <ul style="list-style-type: none"> — AOC—NetView AOC CLIST member — ASM—Assembler program member — CLS—CLIST member — CMD—Command rule — DOC—Documentation member — DOM - Message deletion rule — EOJ - End of job rule — HLL - High level language member — JCL - JCL member — MSG - Message rule — OMG - OMEGAMON exception rule — OPX - OPS/REXX member — REQ - Explicit request member — REX - REXX member — SCR - Screen event rule — SEC - Security exception rule — TOD - Time of day rule — UNK - Unknown product or event type member — VAR - Change of a Global variable rule ■ <SUMDETAIL>--A string containing information about the rule (for example, message ID for a MSG rule, or job name for a end-of-job rule). The string resolved from this keyword is a maximum of eight characters in length. ■ <RUNID>--Date and time the current run of utility CTOFANCI started. This value resolves to a string that is unique for each run of the utility. Therefore, this value can be used to differentiate between the output of successive runs of the utility. The resolution of this keyword is a string with the following format: <i>Dmdd.Thhmm</i> where <i>mdd</i> represents the month and the day, and <i>hhmm</i> represents hours and minutes. <p>The string that results from resolution of the specified keywords must be a valid dataset name.</p>

Table 12 Naming Convention and Location Parameters Created by Utility CTOFANCI (part 3 of 4)

Parameter	Description
OUTDSN (continued)	<p>Examples:</p> <ul style="list-style-type: none"> ■ OUTDSN=IOA.<INDSN> The output dataset name is the input dataset name prefixed by IOA. ■ OUTDSN=CTOP.RULES.<INQ3>.<ETYPE> The output dataset name is CTOP.RULES. followed by the third qualifier of the input dataset name, a period, and the event type code. <p>OUTDSN=CTOP.ALL.RULES All input rules are placed in output dataset CTOP.ALL.RULES.</p>
OUTMEMNAME	<p>Specifies values that determine the name of the member to which each output member or rule must be written. Mandatory.</p> <p>The value specified for this parameter can be a maximum of 72 characters. The value may include any combination of constants (alphanumeric characters) and keywords. Keywords valid for this parameter are the same as those valid for parameter OUTDSN (described elsewhere in this table).</p> <p>The string that results from resolution of the specified keywords must be a valid member name.</p> <p>If the intended name of an output member matches the name of a member already existing in the output library, utility CTOFANCI either appends the output to the existing member, or creates a new member. For more information, see “Output Member Names” on page 42.</p>
OUTVOL	<p>Volser to which new output libraries must be allocated. Optional. A maximum of six characters can be specified.</p> <p>If no value is specified for this parameter, allocation is performed without a volume specification (meaning, the location of output libraries is determined by the default settings of your site’s disk management software).</p>
OUTUNIT	<p>Unit name to which new output libraries must be allocated. Optional. A maximum of eight characters can be specified.</p> <p>If no value is specified for this parameter, allocation is performed without a unit specification.</p>
OUTTRKS1	<p>Specifies the number of primary allocation tracks to be used for allocating new output libraries. Optional. A value from 1 to 9999 can be specified. Default: 10.</p>

Table 12 Naming Convention and Location Parameters Created by Utility CTOFANCI (part 4 of 4)

Parameter	Description
OUTTRKS2	Specifies the number of secondary allocation tracks to be used for allocating new output libraries. A value from 1 to 9999 can be specified. Default: 20
OUTDIR	Specifies the number of directory-blocks to be used for new output libraries. A value from 1 to 9999 can be specified. Default: 100
RLSE	Indicates whether unused space in output libraries must be freed after the current run of utility CTOFANCI. Valid values are: <ul style="list-style-type: none"> ■ Y (Yes) – Free unused space. Default. ■ N (No) – Do not free unused space.
RESET	Indicates what must be done if a library (PDS) already existed with the name of a specified output library prior to this run of utility CTOFANCI. Optional. The existing library was probably created by a previous run of the utility. This parameter determines how to arrange the results of different runs of utility CTOFANCI. Valid values: <ul style="list-style-type: none"> ■ OUTLIB—Create a new library with a different name consisting of desired output library name and a suffix with the value of <RUNID> (described elsewhere in this table). Information is placed in the new empty library instead of the originally specified output library. ■ NONE—The new rule or member is added to the existing output library. Default.
OUTBLKSZ	Block size to be used when creating new output libraries. Optional. A value from 80 to 32760 can be specified. Default: 6000
DUPPREF	Four-character prefix to be used to name each new output member whose intended name matches an existing member's name in the output dataset. Default: @DUP. Utility CTOFANCI appends a numeric suffix (from 0001 to 9999) to this prefix to create a new member name that does not conflict with existing members. The intended output name is included in the DESCRIPTION field of the output rule.

Report Allocation Parameters

The following parameters indicate the location to which reports produced by utility CTOFANCI are written. For more information about these reports, see “CTOFANCI Reports” on page 60

Table 13 Report Allocation Parameters (part 1 of 2)

Parameter	Description
SYSOUT	A one-letter SYSOUT class for reports that are sent directly to a print file and not to a permanent dataset. Optional. Default: X
RPTSRCDSN	<p>Location to which the rule source report (RPTSRC) must be written. Optional. Valid values are:</p> <ul style="list-style-type: none"> ■ <i>dsname</i>—Dataset name of a cataloged sequential file to which the report must be written. ■ DUMMY—No report is created. Specifying this value may enhance the performance time of utility CTOFANCI. <p>If no value is specified for parameter RPTSRCDSN, the report is written to a print file.</p>
RPTRULESDSN	Dataset name of a cataloged sequential file to which the rule report (RPTRULES) must be written. If no dataset name is specified, the report is written to a print file.
RPTRULESMOD	<p>Indicates whether to replace the data in the existing dataset specified in RPTRULESDSN, or to append the new information to the existing data. This parameter indicates how the RPTRULES report must be written if the specified output dataset contains data from a previous run of utility CTOFANCI. Valid values are:</p> <ul style="list-style-type: none"> ■ Y (Yes)—Append new information to the existing report. ■ N (No)—Start writing at the beginning of the dataset. If the dataset already contained information, this information is overwritten. Default. <p>Note: Report RPTRULES is always allocated with an attribute of DISP=SHR (even if RPTRULESMOD=Y was specified). Therefore, you can browse this report while the utility is still running, regardless of the value specified for parameter RPTRULESMOD.</p> <p>This parameter can be used together with the Execution and Restart parameters (described elsewhere in this table) to manage accumulation of the results of multiple runs of utility CTOFANCI.</p>

Table 13 Report Allocation Parameters (part 2 of 2)

Parameter	Description
RPTSRCMOD	<p>Indicates whether to replace the data in the existing dataset specified in RPTSRCDSN, or to append the new information to the existing data. This parameter indicates how the RPTSRC report must be written if the specified output dataset contains data from a previous run of utility CTOFANCI. Valid values are:</p> <ul style="list-style-type: none"> ■ Y (Yes)—Append new information to the existing report. Utility CTOFANCI allocates the report file with DISP=MOD specified. This report can be browsed only after the current run of utility CTOFANCI. ■ N (No)—Start writing at the beginning of the dataset. If the dataset already contained information, this information is overwritten. Default. Utility CTOFANCI allocates the file with DISP=SHR. <p>This parameter can be used together with the Execution and Restart parameters (described elsewhere in this table) to manage accumulation of the results of multiple runs of utility CTOFANCI.</p>

Execution and Restart Parameters

The following parameters can be used to influence basic utility operations, and to indicate how utility CTOFANCI must be run if the previous run of the utility ended due to an error, or because the specified maximum number of input members was processed.

Table 14 Execution and Restart Parameters (part 1 of 4)

Parameter	Description
MODE	<p>Specifies whether output rules must be created by this run of Utility CTOFANCI. Valid values are:</p> <ul style="list-style-type: none"> ■ NORMAL – Utility CTOFANCI creates output libraries, output rules and reports ■ SIMULATION – The utility produces reports but does not create output PDSs or output rules. Reports produced by utility CTOFANCI in simulation mode contain names of output libraries and output rules according to the pattern specified using parameters OUTDSN and OUTTABLE. However, the reports do not reflect any changes in the names of output libraries (PDSs) or output rules that could result from runtime exceptions (for example, duplicate member names, or duplicate library names).

Table 14 Execution and Restart Parameters (part 2 of 4)

Parameter	Description
RESTARTDS	<p>Dataset at which to start processing.</p> <p>If a previous run of utility CTOFANCI ended due to errors or because the specified maximum number of input members was processed, you can use this parameter to indicate that processing must start at a specific dataset and in this way avoid repetition of processing already performed in the previous run of the utility.</p> <p>If no value is specified for parameter RPTSRCDSN, the report is written to a print file.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ■ <i>dsname</i> – Name of the dataset at which to start processing. ■ <i>dsid</i> – Dataset ID (for example, D007) at which to start processing. A dataset ID is assigned by utility CTOFANCI to each input dataset after it is selected using specified INCLUDE statements (see “INCLUDE Statements” on page 57). To determine the dataset ID of a dataset, see report RPTFILES produced by a previous run of the utility. <p>Note: The dataset ID assigned to a specific input dataset may change in subsequent runs of the utility if:</p> <ul style="list-style-type: none"> ■ The INCLUDE statements changed. ■ Certain previously selected input datasets are no longer cataloged. ■ New datasets that match existing INCLUDE statements have been cataloged. <p>Therefore, a dataset ID must be used to indicate restart criteria only if no changes have been made to the selection statements or to your environment since the last run of CTOFANCI.</p> <p>When a value is specified for parameter RESTARTDS, utility CTOFANCI selects all input datasets according to the specified INCLUDE statements, and then skips all selected datasets until the specified one is encountered.</p> <p>Note: You can also use this parameter to skip processing of an input dataset that causes utility CTOFANCI to fail or enter a loop.</p>

Table 14 Execution and Restart Parameters (part 3 of 4)

Parameter	Description
RESTARTMEM	<p>Member at which processing must start.</p> <ul style="list-style-type: none"> ■ If this parameter is specified, utility CTOFANCI skips all members in the first processed dataset, until the member name specified in RESTARTMEM is encountered. ■ If a value has been specified for parameter RESTARTDS, the utility searches for the specified member in the specified dataset. <p>If no value has been specified for parameter RESTARTDS, the utility searches for the specified member in the first dataset selected using the INCLUDE statements.</p> <p>Note: If the specified member is not found in the dataset, the CTOFANCI utility does not process any member in the dataset; instead, it starts processing with the first member of the next dataset. For additional information about restarting the utility in the see the description of the RESTARTDS parameter in this table.</p> <p>You can also use this parameter to skip processing of an input member that causes utility CTOFANCI to fail or enter a loop.</p>
RULEPROGRESS	<p>Number of input rules or members processed between progress messages. Utility CTOFANCI issues a progress message (CTOF28I) each time it finishes processing the specified number of rules and members. Default: 10. For more information about progress message CTOF28I, see Chapter C, "Messages."</p>
SIMLINELIM	<p>Specifies how many lines to process in each input member while in simulation mode. In simulation mode, the utility does not translate automation statements in input members. Therefore, this parameter can reduce resource consumption and improve the performance of the utility in simulation mode. A value from 10 to 999 can be specified.</p> <p>This parameter is ignored if MODE=NORMAL is specified.</p> <p>If an input member contains more than the specified number of lines, the number of lines listed in the utility reports is specified as TURN (that is, truncated).</p> <p>If this parameter is not specified, all lines of each input member are processed.</p> <p>Note: If a truncated member contains multiple rules, some rules may not appear in the reports produced by the simulation run of the utility.</p>

Table 14 Execution and Restart Parameters (part 4 of 4)

Parameter	Description
RULEWARNLIM	<p>Specifies a limit for the number of warnings to be displayed per input rule or member. Optional. A value from 1 to 99 can be specified.</p> <p>Utility CTOFANCI reports problems with input rules or members using warning messages. Certain recurring problems may result in multiple warning messages with similar content.</p> <p>Parameter RULEWARNLIM can be used to reduce the number of warning messages to a manageable number. If messages for a specific rule or member are suppressed as a result of the specified limit, a message is issued indicating the member name and the number of suppressed messages.</p>
STOPAFTER	<p>Specifies a maximum number of input members to be processed in this run of utility CTOFANCI. If the specified number of input members is processed, the utility shuts down.</p> <p>This parameter can be used together with Restart parameters (described elsewhere in this table) to manage accumulation of the results of multiple runs of utility CTOFANCI.</p> <p>If no value is specified for this parameter, all selected input members are processed during this run of utility CTOFANCI.</p>

Translation Parameters

The following parameters specify how to convert certain content in the input members, and how to determine values for certain parameters in output rules.

Table 15 Translation Parameters (part 1 of 2)

Parameter	Description
OUTCOMMENTS	<p>Indicates whether to copy comment lines from the input member to output rules. Optional.</p> <ul style="list-style-type: none"> ■ Y (Yes)—Comments and code are copied from the input rule to the output rule. ■ N (No)—Only code is copied from the input rule to the output rule. Default.
LINESEQ	<p>Specifies the existence and position of line sequence numbers in input scripts:</p> <ul style="list-style-type: none"> ■ AUTO – Utility CTOFANCI checks the content of each input member for presence of sequence numbers. Default. ■ STD – The input members contain sequence numbers in columns 73–80. ■ NONE – The input members do not contain sequence numbers. ■ col1-col2 – The input members contain sequence numbers in the specified line positions (columns). ■ It is recommended that AUTO be specified even though the other values may provide some performance gains. The other values must be used in unusual situations where utility CTOFANCI (with LINESEQ=AUTO) does not properly detect the presence and location of sequence numbers.
RULEMODE	<p>Indicates the operation mode to be specified in Control-O rules created by the utility. Valid values are:</p> <ul style="list-style-type: none"> ■ PROD – Production mode. ■ LOG – Log mode. Default. ■ TEST – Test mode. ■ RULEOWNER – Specifies the user ID to be associated with rules created by utility CTOFANCI. The value specified for this parameter is inserted in the OWNER field of the Control-O rules generated by this run of the utility. Up to 8 characters can be specified. Default: CTOFANCI.

Table 15 Translation Parameters (part 2 of 2)

Parameter	Description
DAYTIME	<p>The start time of the work day at your site. The format is:</p> <p>DAYTIME=+hhmm or DAYTIME=-hhmm</p> <p>where:</p> <ul style="list-style-type: none"> ■ + indicates a time after midnight ■ - indicates a time before midnight ■ hhmm indicates the time (hour and minute) <p>Note: The syntax for this parameter is identical to the syntax for Control-O and Control-M installation parameter DAYTIME. For more information, see the <i>INCONTROL for z/OS Installation Guide</i>.</p> <p>The DAYTIME value must be the same as that assigned to Control-O installation parameter DAYTIME.</p>
SWEEK	<p>Day on which the work week starts. Valid values:</p> <ul style="list-style-type: none"> ■ MON—Start the work week on Monday. Default. ■ SUN—Start the work week on Sunday.

Sample TYPERRUN Statements

The following TYPERRUN statements contain sample subsets of parameters that can be specified to utility CTOFANCI:

Example 1

```
TYPERRUN  OUTDSN=CTO.<INDSN>,OUTMEM=<INMEM>,OUTUNIT=3390,OUTVOL=MYVOL1,...
```

One output dataset is created for each input dataset. The name of the output dataset is the same as the name of the corresponding input dataset, with a prefix of CTO.

Each input member is processed. The resulting member or rule is placed in an output member with the same name in the corresponding dataset.

Example 2

```
TYPERRUN  OUTDSN=CTO.NEW.RULES.<ETYPE>,OUTMEM=MAIN,...
```

All rules with the same type of selection criteria (for example, rules triggered by detection of a message) are grouped together in a separate output PDS (library). Each such PDS contains a single member (a Control-O Rule table) called MAIN.

Example 3

```
TYPERUN  OUTDSN=MY.SITE.RULES,OUTMEM=<INMEM>...
```

All output rules are placed in one output library. Output member names are the same as the names of the corresponding input members.

INCLUDE Statements

INCLUDE statements specify selection criteria for members to be processed by the utility. Multiple INCLUDE statements can be specified.

The format of an INCLUDE statement is as follows:

```
INCLUDE DSN=dsname,MEMBER=memname
```

where:

- *dsname* is the dataset name (or mask) of the datasets to be processed by the utility.
- *memname* is the member name (or mask) of the members to be processed by the utility.

Each specified value must be either a full name (dataset name or member name) or a valid mask (that is, a name prefix followed by an asterisk). An asterisk can appear in a dataset name mask only after a period (that is, you must specify complete qualifiers).

Any number of INCLUDE statements can be specified. Multiple INCLUDE statements can overlap (that is, a given dataset can match the selection criteria in more than one INCLUDE statement). INCLUDE statements that specify similar criteria can be used to select multiple sets of members from the same dataset. Each member selected using INCLUDE criteria is processed only once, regardless of the number of INCLUDE statements that match the member.

Sample INCLUDE Statements

```
INCLUDE DSN=SITE1.MVS1.OPS.RULES.*,MEMBER=*
INCLUDE DSN=SITE2.MESSAGES.*
INCLUDE DSN=SITE3.MVS2.AUTOMATE.*,MEMBER=MSG*
```

**NOTE**

EXCLUDE statements (specifying members not to be processed) are not yet supported by utility CTOFANCI.

JCL Required to Execute CTOFANCI

The following is a sample of the JCL statements that activate utility CTOFANCI. For more information about activation of this utility, see [Chapter 2, “Conversion Steps.”](#)

Figure 2 JCL Required to Execute CTOFANCI

```
//CTOFANCI JOB      ,NAME,MSGCLASS=X,CLASS=A,NOTIFY=NAME
//CTOFANCI EXEC    PGM=IKJEFT01,DYNAMNBR=999,REGION=4M
//SYSEXEC DD DISP=SHR,DSN=CTOF.REXX
//          DD DISP=SHR,DSN=SYS2.EXEC
//SYSPROC DD DISP=SHR,DSN=CTOF.REXX
//          DD DISP=SHR,DSN=SYS2.CLIST
//SYSPRINT DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
PROF NOPREFIX
CTOFANCI CTOF.REXX(SYSINPRM)
```

Reports

This chapter includes the following topics:

CTOFANCI Reports.	60
RPTALL – General Message Report	60
RPTWARN – Warning Message Report.	61
RPTFILES – Input/Output File Report.	61
RPTRULES – Input Rule Report	62
RPTSRC – Input Rule Source Report	65
Title Lines	65
Text Lines	66
Sort-Stubs in Report RPTSRC	66

CTOFANCI Reports

A number of reports are generated by each run of utility CTOFANCI. Each report contains information describing certain aspects of your current console management configuration and how it was handled by utility CTOFANCI. This information can help you plan and subsequently manage your conversion effort.

Each CTOFANCI report is produced in a file referenced by a separate DD statement. The following reports are generated:

Table 16 CTOFANCI Reports

Report	Description
RPTALL	General Message Report
RPTWARN	Warning Message Report
RPTFILES	Input/Output File Report
RPTRULES	Rule Report
RPTSRC	Rule Source Report

Each of these reports is described in detail below.

NOTE



Reports RPTRULES and RPTSRC have a fixed format. You can view these reports as generated, or redirect them to a disk and perform your own sorting and reporting from the disk files.

If a unit of information produced by the utility cannot fit in a fixed-size report field, an overflow occurs and the data is replaced with the symbol “+”.

A special layout member describes the format of each report (for an explanation, see the following section).

RPTALL – General Message Report

Report RPTALL contains a copy of the control statements specified for utility CTOFANCI (that is, TYPERRUN and INCLUDE statements) and all messages that were generated by the utility.

Messages generated by utility CTOFANCI are described in detail in [Chapter C, “Messages.”](#)

RPTWARN – Warning Message Report

Report RPTWARN contains all warning, error, and severe error messages issued by utility CTOFANCI. The messages in this report are a subset of the messages listed in report RPTALL.

Messages generated by utility CTOFANCI are described in detail in [Chapter C](#), “Messages.”

This report can be used to review problems encountered by utility CTOFANCI and to determine what manual adjustments are necessary.

RPTFILES – Input/Output File Report

This report describes all input and output datasets processed or created by utility CTOFANCI. The report also indicates the dataset ID that was assigned to each input dataset name.

The following fields are included in report RPTFILES:

Table 17 Fields in RPTFILES Report (part 1 of 2)

Field	Description
DSN ID	<p>Dataset ID of the dataset. Each dataset processed or created by utility CTOFANCI is assigned an ID number. This dataset ID is used in place of the complete dataset name in other reports produced by the utility.</p> <ul style="list-style-type: none"> ■ Dataset IDs of input datasets are numbered D001, D002, and so on. ■ Dataset IDs of output datasets are numbered D101, D102, and so on. <p>If more than 100 datasets are selected as input for the utility, output dataset IDs are assigned numbers starting with the lowest multiple of 100 not already assigned to an input dataset.</p>
IN/OUT	<p>Whether the file described in this line is an input file or an output file. Possible values are:</p> <ul style="list-style-type: none"> ■ IN—Input file. ■ OUT—Output file.

Table 17 Fields in RPTFILES Report (part 2 of 2)

Field	Description
DATASET STATUS	Indication of how the dataset was handled by utility CTOFANCI. Valid values: <ul style="list-style-type: none"> ■ UNCHANGED – This value is assigned to input datasets. These datasets are not modified by utility CTOFANCI. ■ NEW – The dataset was created by this run of utility CTOFANCI. ■ UPDATED – The dataset existed prior to, but was modified during, this run of the utility.
DATASET NAME	Complete dataset name.
MEMBERS	Member names or masks specified in INCLUDE statements of the utility, which were used to select the dataset. This field is not relevant for output datasets.

The following report segment shows a sample of RPTFILES output.

Figure 3 Sample RPTFILES Output

```

-----
CTOFANCI: DATASET INDEX REPORT (RPTFILES) RUN ID = D1027.T1016
-----
DSN  IN/ DATASET
ID   OUT STATUS  DATASET NAME                MEMBERS
-----
D001 IN  UNCHANGED  CTO.RULES.TEST.SPECIAL     SPCLA2LN
D101 OUT NEW      CTO.OUTPUT                  -
    
```

RPTRULES – Input Rule Report

Report RPTRULES describes input rules or members that were processed by the utility. Each line of the report describes one input rule regardless of whether it was the only rule in the input member.

- Many input members contain only one rule. Each of these members is described in a single line of report RPTRULES.
- Sometimes multiple rules are defined in a single input member. Each of these rules is described in a separate line of report RPTRULES.

The following fields are included in report RPTRULES:

Table 18 Fields in the RPTRULES Report (part 1 of 2)

Field	Description
IN DSN ID	Dataset ID of the PDS in which the member containing the input rule was found. For more information about dataset IDs, see “ RPTFILES – Input/Output File Report ” on page 61.
MEMBER NAME	Input member name.
WARN	Warning indicator field. This field contains an asterisk (*) if a warning message was issued while processing the rule described in this line of the report. If no warning was issued, this field is blank.
INPUT RULE SEQ NUM	Unique sequence number of the input rule within the input member (R0001, R0002, and so on.).
TOTAL MEM	Number of lines in the input member (including all rules in the member). The same value appears for all the rules in the same input member.
SCRIPT	Number of lines (in the input member) that contain text that is not a comment (for example, /* comment */), sequence number (columns 73 - 80 in numbered FB files or columns 1 - 8 in numbered VB files), or spaces. The same value appears for all the rules in the same input member.
RULE	Number of lines of code in the rule (excluding comment lines and blank lines).
PRODUCT NAME	The product (or the language) associated with the rule. Utility CTOFANCI currently recognizes the following types of member content: <ul style="list-style-type: none"> ■ CA-OPS ■ CA-AUTOMATE ■ MVS Message Processing facility (MPF) ■ Native TSO REXX ■ MVS JCL ■ Documentation ■ Native TSO CLIST ■ NetView AOC ■ High level language (for example, C or PL/I) ■ Program in Assembler 370 or 390 ■ Control-O Rule ■ PDSMAN(\$\$\$\$SPACE members)

Table 18 Fields in the RPTRULES Report (part 2 of 2)

Field	Description
EVENT TYPE CODE	<p>The event type code of the input member or rule. The utility automatically assigns one of the supported event type codes to each input rule or member. Input members that do not contain rules are assigned an event type code according to the product or language of the member. The following event type codes are used:</p> <ul style="list-style-type: none"> ■ AOC—NetView AOC CLIST member ■ ASM—Assembler member ■ CLS—CLIST member ■ CMD—Command rule ■ DOC—Documentation member ■ DOM—Message deletion event rule ■ EOJ—End of job rule ■ HLL—High level language member ■ JCL—JCL member ■ MSG—Message rule ■ OMG—OMEGAMON exception rule ■ OPX—OPS/REXX member ■ REQ—Explicit request member ■ REX—REXX member ■ SCR—Screen event rule ■ SEC—Security exception rule ■ TOD—Time of day rule ■ UNK—Unknown event type ■ VAR—Change of a Global variable rule
OUT DSN ID	Dataset ID of the output dataset to which the corresponding output rule or member is written.
OUTPUT TABLE/MEMBER	Member or Rule Table name to which the corresponding output rule or member is written.
INTENDED OUTPUT MEMBER	<p>Name of the member or Rule table to which the corresponding output rule must have been written (for rules or members that were assigned a duplicate name (@DUPnnnn).</p> <p>If the rule was actually written to the intended member, the Output Table or Member and Intended Output Table or Member fields contain the same value.</p>
FULL DETAILS	Information about the event that triggers the rule (such as, message ID, time specification, name of a Global variable.).
DETAILS SUMMARY	A subset of the rule details supplied in the Detail field (for example, message ID, jobname).

The following screen segment contains part of a RPTRULES report. Note that this report contains additional columns not shown in this sample.

Figure 4 Sample RPTRULE Report

```

-----
CTOFANCI:  INPUT RULE SUMMARY REPORT (RPTRULES) RUN ID = D1027.T1256
-----

```

IN		W INPUT	NUM-OF-LINES			EVENT		OUT	OUTPUT
DSN	MEMBER	A RULE	TOTL	SCR-	PRODUCT	TYPE	DETAILS	DSN	TABLE/
ID	NAME	N NUM	MEM	IPT	RULE	NAME	SUMMARY	ID	MEMBER
D001	\$\$\$\$SPACE	R0001	0	0	0	PDSMAN	-		
D001	C10ERR1	R0001	66	12	12	CLIST	CLIST	D101	MAIN
D001	DBASUN	R0001	4	1	1	UNKNOWN	UNKNOWN	D101	@DPL0001
D001	IEF085I	R0001	10	1	1	CLIST	CLIST	D101	@DPL0002
D001	INCBKUP	R0001	80	67	67	CLIST	CLIST	D101	@DPL0003

RPTSRC – Input Rule Source Report

Report RPTSRC contains the source lines of the input rules.

Each line of report RPTSRC contains 80 bytes of source code followed by a “sort-stub” (described below) that can be used to sort the report.

The lines of report RPTSRC are divided into title lines and text lines.

Title Lines

Title lines are provided in sets of three at the beginning of each rule in report RPTSRC.

The first and third header lines contain only dashes. The second line of the header contains 80 bytes of rule information extracted from the rule report (RPTRULES).

The following screen segment contains sample title lines (without the sort-stub):

Figure 5 RPTSRC Input Rule Source Report Title Lines

```

-----
D001 C10ERR1      R0001   66   12   12 CLIST   CLIST                D101 MAIN   ...
-----

```

Each set of title lines is assigned line numbers A0001, A0002, and A0003 (in the sort-stub).

Text Lines

Text lines in report RPTSRC contain the text of rules selected as input for utility CTOFANCI. Text lines contain the original script lines without comments and without extra blanks.

The following screen segment contains sample text lines (without the sort-stub):

Figure 6 RPTSRC Input Rule Source Report Text Lines

```
PROC 1 SITE
  GOTO MESSAGE
LINE1: +
  SET &CIANUM = 1
  GOTO MESSAGE
LINE2: +
  SET &CIANUM = 2
  GOTO MESSAGE
```

Text lines are assigned line numbers beginning in the letter B in the sort-stub (for example, B0001, B0002, B0003).

Sort-Stubs in Report RPTSRC

A sort-stub is a string containing information that can be used to sort lines in report RPTSRC. Each record in the report RPTSRC contains a sort-stub (beginning in column 81). Information in the sort stub includes:

- Dataset ID
- Member name
- Rule-sequence-number
- Line number
- Number of lines in member
- Number of lines in the member not counting comments and blank lines
- Number of lines in rule
- Product code
- Event type code
- Output dataset ID
- Output member name

Rules can be sorted into groups to suit your conversion work plan, while keeping all the lines of a given rule together.

The following screen segment contains sort-stubs:

Figure 7 RPTSRC Input Rule Source Report Sort-Stubs

D001	C10ERR1	R0001	A0001	66	12	12	CLS	CLS	D101	MAIN
D001	C10ERR1	R0001	A0002	66	12	12	CLS	CLS	D101	MAIN
D001	C10ERR1	R0001	A0003	66	12	12	CLS	CLS	D101	MAIN
D001	C10ERR1	R0001	B0001	66	12	12	CLS	CLS	D101	MAIN
D001	C10ERR1	R0001	B0002	66	12	12	CLS	CLS	D101	MAIN
D001	C10ERR1	R0001	B0003	66	12	12	CLS	CLS	D101	MAIN

NOTE

When sorting report RPTSRC, the last four sort fields must be Dataset ID, Member name, Rule-sequence-number, and Line number (in that order). Including these sort fields in this way, ensures that each rule is sorted as a unit (meaning, title and text lines of each rule are kept together and in order).

Troubleshooting

The CTOFANCI utility analyzes text in input scripts (rules) to determine how, or whether, to handle them during the conversion process.

While most types of input are properly handled by utility CTOFANCI, an unexpected combination of conditions can cause utility CTOFANCI to fail. Use the information provided below to handle CTOFANCI failures and to minimize their effect on your automation conversion effort.

Bypassing CTOFANCI Problems

If the CTOFANCI utility ended abnormally due to problems with a specific input dataset or member, you can skip the problematic dataset or member using parameters `RESTARTDS` or `RESTARTMEM` (described in [Chapter 4, “Utility CTOFANCI”](#)).

Analyzing the order of files in report `RPTFILES` and the order of member patterns for each file can locate the problematic member. In most cases, the problematic member is the one selected immediately following the last member listed in report `RPTSRC`.

Preventing Infinite Loops

Use the following methods to prevent infinite loops during a run of utility CTOFANCI:

- Specify a time limit in the JCL JOB statement that is used to activate utility CTOFANCI.
- Manually monitor execution of the conversion utility.

Under normal circumstances, new rules are added to the RPTRULES report every few seconds. The number of rules that can be observed at runtime depends on the block size of the file to which the report is being written. If the RPTRULES report is not updated at the expected frequency, cancel execution of utility CTOFANCI.

- Manually monitor the messages produced by the utility.

Utility CTOFANCI issues progress message CTOF28I each time a specified number of input rules and/or members are processed. If this message stops appearing at the expected frequency, cancel execution of utility CTOFANCI.

If an infinite loop occurs, cancel the job and use the instructions in Appendix B to report the problem to your BMC Software Customer Support representative.

Problem Reporting

Provide all the information requested below when reporting a problem related to conversion to Control-O:

- Type of MVS operating system (for example, MVS version 5.2).
- Name and version number of the console automation products from which you are converting.
- Tape number and version of the Conversion Utility (for example, FD1111, level 9710).
- Full details of all error messages that were generated.
- All Job Log messages.
- A copy of the control statements that were used to invoke utility COTFANCI (that is, the TYPERSUN and INCLUDE statements).
- A copy of the report produced by utility CTOFANCI on the file referenced by DD statement RPTALL.
- A copy of the input lines (or entire script) that caused the conversion to fail.
- If the tool enters an infinite loop, follow the procedure outlined in “Reporting Infinite Loops” (below).
- If an ABEND occurs, send the symptom dump to BMC Software Customer Support and have a full dump (SYSUDUMP) available in case the BMC Software Support staff requests it.

Reporting Infinite Loops

To help the BMC Software, Inc. development staff diagnose suspected infinite loops, rerun utility CTOFANCI in TSO (instead of running it in batch) using the following steps:

- 1** Examine the batch JCL of the utility to determine which datasets it uses.
- 2** Reference the datasets identified in step 1 using DD statements SYSPROC, SYSEXEC or STEPLIB in your logon procedure.
- 3** Logon to TSO, and enter ISPF (Option 6).
- 4** Type `CTOFANCI sysin-dsn(sysin-member)` as entered in the batch run.
- 5** Monitor the TSO execution of utility CTOFANCI (from a different TSO session) until an infinite loop situation or another type of problem is suspected.
- 6** Press the TSO Attention key in the TSO session running utility CTOFANCI. A REXX prompt is displayed.
- 7** Type TS (meaning, TRACE START) in response to the REXX prompt, and press <Enter>.
- 8** Print the screen, or capture its content in a file.
- 9** Continue pressing <Enter> and capturing the screen content several times (for example, ten times).
- 10** Press the TSO Attention again. A REXX prompt is displayed.
- 11** Respond HI (meaning, Halt Interpretation) in response to the REXX prompt. Utility CTOFANCI is stopped.

Messages

This list contains all messages produced by conversion utility CTOFANCI. A message code usually consists of the following parts:

*CTOFnn*s

where:

- *nn* is a 2 character numeric identifier for the message.
- *s* is a 1 character code identifying the severity of the message:
 - E—Error
 - S—Severe
 - W—Warning
 - I—Information

CTOF00I **PROCESSING STARTED** *time*

Explanation: This information message indicates the time when the CTOFANCI utility started processing.

User response: You can use the message time to help monitor the performance of CTOFANCI.

CTOF02E **UNRECOGNIZED CONTROL STATEMENT** *word1*

Explanation: An invalid control statement was specified to utility CTOFANCI.

word1 is the first word in the invalid control statement.

Control statements specified for utility CTOFANCI must begin with either INCLUDE or TYPERUN.

System action: Utility CTOFANCI stops execution.

User response: Correct the invalid control statement and rerun utility CTOFANCI.

CTOF03E INVALID KEYWORD IN CONTROL STATEMENT *keyword*

Explanation: An invalid parameter name (keyword) was specified for utility CTOFANCI.

Valid parameters are listed in [Chapter 4, “Utility CTOFANCI.”](#)

System action: Utility CTOFANCI stops execution.

User response: Correct the invalid control statements and rerun utility CTOFANCI.

CTOF04E INVALID DATASET NAME SPECIFICATION *dsn*

Explanation: A dataset name specified in an INCLUDE statement for utility CTOFANCI is invalid or could not be found.

dsn is the invalid dataset name.

System action: Utility CTOFANCI stops execution.

User response: Correct the input dataset specification and rerun utility CTOFANCI.

CTOF05E INVALID MEMBER NAME SPECIFICATION *memname*

Explanation: An invalid member name or mask was specified in an INCLUDE statement for utility CTOFANCI.

memname is the invalid member name or mask.

System action: Utility CTOFANCI stops execution.

User response: Correct the input member name or mask specification and rerun utility CTOFANCI.

CTOF06E INVALID VALUE SPECIFIED *parm=value*

Explanation: An invalid value was specified for a parameter in the control statements of utility CTOFANCI.

- *parm* is the name of the parameter for which the invalid value was specified.
- *value* is the invalid value that was specified.

Valid parameter values are listed in [Chapter 4, “Utility CTOFANCI.”](#)

System action: Utility CTOFANCI stops execution.

User response: Correct the parameter specification and rerun utility CTOFANCI.

CTOF07E**PARAMETER VALUE TOO LONG *parm=value***

Explanation: A value that is too long was specified for a parameter of utility CTOFANCI.

- *parm* is the name of the parameter for which the invalid value was specified.
- *value* is the invalid value that was specified.

Valid parameter values are listed in [Chapter 4, “Utility CTOFANCI.”](#)

System action: Utility CTOFANCI stops execution.

User response: Correct the parameter specification and rerun utility CTOFANCI.

CTOF08E**PARAMETER VALUE OUT OF RANGE *parm=value***

Explanation: A value outside of the permitted range was specified for a parameter of utility CTOFANCI.

- *parm* is the name of the parameter for which the invalid value was specified.
- *value* is the invalid value that was specified.

Valid parameter values are listed in [Chapter 4, “Utility CTOFANCI.”](#)

System action: Utility CTOFANCI stops execution.

User response: Correct the parameter specification and rerun utility CTOFANCI.

CTOF09E**DATASET *dsn* IS NOT A PARTITIONED DATASET**

Explanation: The specified dataset is not a partitioned dataset (PDS).

Only partitioned datasets are processed by utility CTOFANCI (as input or output).

This problem can occur if an output rule library was pre-allocated without any directory blocks. The resulting dataset is sequential and cannot be used by utility CTOFANCI.

System action: Utility CTOFANCI stops execution.

User response: Specify a different dataset, or recreate the dataset as a PDS, and rerun utility CTOFANCI.

CTOF10E**DATASET NOT FOUND DD=*ddname* DSN=*dsn***

Explanation: The specified dataset could not be found.

This problem can occur if you did not preallocate the datasets specified using parameters RPTRULESDSN or RPTSRCDSN.

System action: Utility CTOFANCI stops execution.

User response: Specify a different dataset, or create the missing dataset, and rerun utility CTOFANCI.

CTOF11E **GENERATED DSN *dsn* TOO LONG**

Explanation: The resolved name of an output dataset to be created by utility CTOFANCI is more than 44 characters in length.

Output dataset names are resolved from the name pattern specified for parameter OUTDSN.

This message may also be issued if RESET=OUTLIB was specified, and the unique suffix added to the dataset name caused it to exceed 44 characters.

System action: Utility CTOFANCI stops execution.

User response: Adjust the OUTDSN and/or RESET parameters and rerun the CTOFANCI utility.

CTOF12E **MANDATORY PARAMETER *parm* MISSING**

Explanation: The specified mandatory parameter was not included in the control statements for utility CTOFANCI.

System action: Utility CTOFANCI stops execution.

User response: Specify a value for the required parameter and rerun utility CTOFANCI.

CTOF13E **PARAMETER SPECIFICATION CONFLICT *parm1=value1 parm2=value2***

Explanation: The specified parameters and values conflict with each other.

System action: Utility CTOFANCI stops execution.

User response: Modify at least one of the parameters and rerun utility CTOFANCI.

CTOF16I **PROCESSING TERMINATED *time***

Explanation: This information message indicates that utility CTOFANCI stopped running due to errors.

The time when the utility terminated is included in this message.

System action: Utility CTOFANCI stops execution.

User response: Check the reports and rules generated by the terminated run of the utility. You can choose to rerun the utility from the beginning, or you can use the restart parameters of the utility to restart processing from the point at which the utility terminated.

CTOF17E **UNABLE TO ALLOCATE OR OPEN DD *ddname* [DATASET *dsn*] [MEMBER *member*]**

Explanation: Utility CTOFANCI was unable to access the specified dataset or member.

This problem can occur if another user is holding the dataset, preventing access by utility CTOFANCI.

System action: Utility CTOFANCI stops execution.

User response: To determine the cause of the failure, see other messages in the console log and in the file referenced by DD statement SYSTSPRT. Correct the problem and rerun utility CTOFANCI.

CTOF18I **MEMBER PROCESSING STARTED. MODE=*umode* *msgtime***

Explanation: This information message indicates that CTOFANCI completed initialization and started processing individual members.

umode is the mode in which the CTOFANCI utility is running (that is, NORMAL or SIMULATION).

msgtime is the time when the message was issued.

User response: The time included in this message can be used to help monitor the performance of utility CTOFANCI.

CTOF19E **EMBEDDED SPACES NOT ALLOWED**

Explanation: A blank was encountered in a parameter specification in a position where blanks are not allowed.

System action: Utility CTOFANCI stops execution.

User response: Correct the invalid parameter specification and rerun utility CTOFANCI.

CTOF20E **INVALID <KEYWORD> USAGE *keyword***

Explanation: An invalid keyword was specified for a parameter of utility CTOFANCI.

Keywords enclosed in angle brackets (<>) can be used to specify output dataset and member name patterns for parameters OUTDSN and OUTMEMNAME.

System action: Utility CTOFANCI stops execution.

User response: Correct the invalid parameter specification and rerun utility CTOFANCI.

CTOF21I **CREATING NEW LIBRARY *alloc-details***

Explanation: This information message indicates that the CTOFANCI utility is creating a new partitioned dataset (PDS).

alloc-details is a string of information about the PDS being created. The PDS name, size, and other allocation parameters (if any) are provided in this string.

System action: The job continues.

CTOF22E **I/O ERROR OR SX37 ABEND ON DDNAME=*ddname***

Explanation: An I/O operation on the file referenced by DD statement *ddname* failed.

This problem can occur if the specified dataset is full.

System action: Utility CTOFANCI terminates.

User response: To determine the cause of the failure, see other messages in the console log and in the file referenced by DD statement SYSTSPRT. Correct the problem and rerun utility CTOFANCI, or restart it from the dataset or member at which processing was halted.

CTOF23E **INVALID SYNTAX OR A CONTINUATION ERROR**

Explanation: An error was encountered while processing the control statements specified for utility CTOFANCI.

Each parameter specification for the utility must appear on a single line. This message may indicate that a parameter specification was continued on a second line. For additional syntax rules, see “[Activating Utility CTOFANCI](#)” on page 43

System action: Utility CTOFANCI stops execution.

User response: Correct the invalid parameter specification and rerun utility CTOFANCI.

CTOF24E **CONTROL STATEMENT IS REDUNDANT**

Explanation: A control statement was encountered more than the allowed number of times.

System action: Utility CTOFANCI stops execution.

User response: Either delete the extra control statements or incorporate them into other control statements, and rerun utility CTOFANCI.

CTOF25E **OUTPUT DATASET CONFLICTS WITH INPUT *dsn***

Explanation: The same dataset name was specified (or generated from a mask) for both input and output of utility CTOFANCI.

To protect existing data, utility CTOFANCI does not write to a dataset whose name is identical to a selected input dataset name.

System action: Utility CTOFANCI stops execution.

User response: Modify the INCLUDE statements and/or the specification for the OUTDSN parameter, and rerun the CTOFANCI utility.

CTOF26E

INVALID OUTPUT DATASET NAME GENERATED *dsn*

Explanation: Utility CTOFANCI attempted to create an output dataset name that does not comply with JCL dataset name specification constraints.

The naming pattern specified for output datasets (using parameter OUTDSN) resolved to an invalid dataset name. This problem can also occur if you specified a dataset name pattern that could not be successfully resolved (for example, OUTDSN=MYPREFIX.<INQ4> was specified, and at least one of the input datasets did not have a 4th qualifier).

System action: Utility CTOFANCI stops execution.

User response: Correct the problem and rerun utility CTOFANCI, or restart the utility from the dataset or member at which processing was halted.

CTOF27E

INVALID OUTPUT TABLE/MEMBER NAME GENERATED *member name*

Explanation: Utility CTOFANCI attempted to create an output member name that does not comply with JCL member name specification constraints.

The naming pattern for output members is specified using parameter OUTMEMNAME. This message may be issued if a resolved output member name was more than eight characters in length.

System action: Utility CTOFANCI stops execution.

User response: Correct the problem and rerun utility CTOFANCI, or restart it from the member at which processing was halted.

CTOF28I

PROCESSED *numrul* RULES, *nummem* MEMBERS [AT: D= *dsid* M=*memname* R=*rulnum*] *time*

Explanation: This information message is issued after a specified number of actions to describe the progress of the current run of utility CTOFANCI.

The message is issued each time a specified number of input members or rules is processed by the utility. This message is also issued when the utility terminates execution.

Each copy of this message include the following information:

numrul: The number of input rules processed thus far in the current run.

nummem: The number of input members processed thus far in the current run.

dsid: Dataset ID of the input dataset currently being processed.

memname: Member name of the input member currently being processed.

rulnum: Sequence number of the rule currently being processed.

time: The time the message was issued.

Members skipped due to restart parameters are not included in the count. If the program continues, the message also includes the name of the next member to be processed.

System action: The utility continues processing.

User response: You can use the information in this message to help monitor the execution of utility CTOFANCI.

CTOF30I

RESTART: SKIPPING ENTIRE DATASET *dataset-id dsn*

Explanation: This information message indicates that the specified input dataset was not processed because of specified restart parameters.

Restart parameter RESTARTDS can be used to indicate a specific dataset at which the CTOFANCI utility begins processing. All datasets selected (according to INCLUDE criteria) prior to the specified dataset are skipped (meaning, they not processed).

This message is issued for each dataset skipped as a result of the specification for parameter RESTARTDS.

System action: Utility CTOFANCI continues processing with the next dataset.

CTOF31I

RESTART: SKIPPING MEMBER *M=member DSID=dataset-id DSN=dsn*

Explanation: This information message indicates that the specified input member was not processed because of specified restart parameters.

Restart parameter RESTARTMEM can be used to indicate a specific member at which the CTOFANCI utility begins processing. All members selected (according to INCLUDE criteria) prior to the specified member are skipped (meaning they not processed).

This message is issued for each member (in the first processed dataset) that is skipped as a result of the specification for parameter RESTARTMEM.

System action: Utility CTOFANCI continues processing with the next member.

CTOF32I**RESTART: PROCESS STARTING AT M=member DSID=dataset-id DSN=dsn**

Explanation: This information message indicates that the CTOFANCI utility located and has begun processing with the specified restart dataset and/or member.

All datasets and members (specified using INCLUDE statements) that are selected prior to the specified restart dataset or member are skipped (meaning they are not processed in this run of utility CTOFANCI).

CTOF32W**M=member D=dsid L=line STRING IN QUOTES (‘’) SPANS MULTIPLE LINES – ACCEPTED**

Explanation: The specified input script contains an unusually long string enclosed in quotes.

The string may be parsed incorrectly by utility CTOFANCI.

System action: Utility CTOFANCI continues processing.

User response: Check the indicated output rule to determine if utility CTOFANCI correctly parsed the text. If the parsing was incorrect, manually adjust the rule as necessary.

CTOF33W**M=member D=dsid L=line COMMENT MARKS (/* */) INSIDE QUOTES – IGNORED**

Explanation: The specified input script contains a comment indicator inside a constant string (for example, WTO ‘THIS IS AN OPERATOR MESSAGE /*’).

The string may be parsed incorrectly by utility CTOFANCI.

System action: Utility CTOFANCI continues processing.

User response: Check the indicated output rule to determine if utility CTOFANCI correctly parsed the text. If the parsing was incorrect, manually adjust the rule as necessary.

CTOF34W**M=member D=dsid L=line COMMENT MARK (/*) INSIDE A COMMENT – IGNORED**

Explanation: The specified input script contains a nested comment (for example, /* A /* B */).

The string may be parsed incorrectly by utility CTOFANCI.

System action: Utility CTOFANCI continues processing.

User response: Check the indicated output rule to determine if utility CTOFANCI correctly parsed the text. If the parsing was incorrect, manually adjust the rule as necessary.

CTOF35W**M=member D=dsid L=line UNMATCHED COMMENT-END MARK (*) - IGNORED**

Explanation: The input script contains an unexpected comment-end mark (for example, A = B + C */).

The string may be parsed incorrectly by utility CTOFANCI.

System action: Utility CTOFANCI continues processing.

User response: Check the indicated output rule to determine if utility CTOFANCI correctly parsed the text. If the parsing was incorrect, manually adjust the rule as necessary.

CTOF36W**M=member D=dsid L=line COMMENT END (*) NOT FOUND**

Explanation: The input script contains a comment that was not properly closed (for example, A = B + C /*).

The string may be parsed incorrectly by utility CTOFANCI.

System action: Utility CTOFANCI continues processing.

User response: Check the indicated output rule to determine if utility CTOFANCI correctly parsed the text. If the parsing was incorrect, manually adjust the rule as necessary.

CTOF37W**M=member D=dsid L=line MATCHING QUOTE ("") NOT FOUND**

Explanation: The input script contains a constant string that was not properly closed (for example, WTO ' THIS STRING IS NOT CLOSED).

The string may be parsed incorrectly by utility CTOFANCI.

System action: Utility CTOFANCI continues processing.

User response: Check the indicated output rule to determine if utility CTOFANCI correctly parsed the text. If the parsing was incorrect, manually adjust the rule as necessary.

CTOF38W**M=member D=dsid L=line ---- MORE WARNINGS FOR MEMBER WERE SUPPRESSED ----**

Explanation: Warning messages were suppressed for the specified input member or rule.

The user specified the maximum number of warning messages to display for input members (using parameter MEMWARNLIM). More than this specified maximum number of warning messages were generated while processing the member indicated in this message.

System action: Utility CTOFANCI suppresses the additional messages, and continues processing.

User response: If you are either uninterested in the additional warnings (for example, because you are aware of the problems they describe), or if you are satisfied with the actual results of the conversion (the output rule), then no action is needed.

Otherwise, increase the value specified for MEMWARNLIM and rerun or restart the utility.

CTOF39E **M=member D=dsid L=line UNABLE TO PARSE TEXT *text* - LINE ACCEPTED**

Explanation: The utility was not able to analyze the input script completely.

This error message indicates that the specified input string could not be handled by the utility for an unspecified reason.

System action: Utility CTOFANCI continues processing.

User response: Check the indicated output rule to determine if utility CTOFANCI correctly parsed the text. If the parsing was incorrect, manually adjust the rule as necessary.

In addition, report the problem to your BMC Software Customer support representative. Attach a copy of the input script and the output messages in the RPTALL report.

TOF40I **REACHED MEMBER-COUNT LIMIT (*number*) PROCESSING STOPPED**

Explanation: This information message indicates that the CTOFANCI utility completed processing a specified maximum number of input members or rules.

Use parameter STOPAFTER to specify the maximum number of input members or rules that must be processed by the current run of the utility.

System action: Utility CTOFANCI stops execution.

User response: Review the reports and rules produced by this run of the utility. If necessary, determine which selected input datasets and members were not processed, and restart the utility from the first dataset and/or member that was not processed in this run.

CTOF41E **PREFIX *duppref=prefix* IS INVALID. MEMBER NAME *member* HAS A NON-NUMERIC SUFFIX**

Explanation: The utility was not able to generate duplicate member names (for example, @DUP0001, @DUP0002) because members whose name begins with the same prefix were found in the output library with a suffix that is not entirely numeric; for example, @DUP9PLX.

When the first duplicate member name is detected, utility CTOFANCI checks the output library for members with the specified duplicate member name prefix. The utility uses this information to determine the number to append to the first duplicate member name assigned during the current run.

If a member with the duplicate name prefix exists with a non-numeric suffix, the utility may be unable to calculate the number to assign to the first duplicate member detected during this run.

System action: Utility CTOFANCI stops execution.

User response: Check the names of members in the output library and do one of the following:

- Rename or remove the problematic members in the output library.
- Specify a different value for parameter DUPPREF of utility CTOFANCI.

CTOF42E

DUPLICATE-MEMBER SUFFIX LIMIT EXCEEDED – *number*

Explanation: Utility CTOFANCI exceeded the maximum number that can be appended to a duplicate-member name.

The highest number that can be added to the duplicate-member prefix is 9999.

Note that the CTOFANCI utility searches for duplicate-member name with the highest number appended to it in the output library, to determine which number to assign to the first duplicate-member name in the current run of the utility. For example, if member @DUP9995 existed in the output library before the beginning of the current run, a maximum of four duplicate-member names can be assigned in this library during this run of the utility.

System action: Utility CTOFANCI stops execution.

User response: Check the member names in the output library.

- If the problem was caused by an unexpected member with the duplicate-member prefix and a high number (for example, there was a single member in the output library named @DUP9995), delete or rename the offending member.
- If the cause of the problem was that the permitted number of members with duplicate names was exceeded, specify a different value for parameter DUPPREF and restart utility CTOFANCI from the next selected dataset or member.

If a large number of duplicate-member names were assigned, it may be preferable to rerun utility CTOFANCI with a more elaborate pattern specified for output member names (parameter OUTMEMNAME).

CTOF43W

INPUT FILE IS EMPTY D=*dsn*

Explanation: The specified input dataset contains no members.

System action: Utility CTOFANCI continues processing with the next selected input dataset.

User response: Verify that the input datasets indicated using the INCLUDE statements are correctly specified.

CTOF45W

M=member D=dsid [L=line] inparm = "invalue" Æ outtext

Explanation: The specified input data was only partially converted to Control-O parameters.

outtext is a string that shows how the input data was translated.

Certain features of other console automation programs are converted in ways that are not identical to the input script. For example, a minute is the smallest unit of time that can be specified in a Control-O rule. If the input script specifies the time 00:00:30 (30 seconds after midnight), the time specified in the output Control-O rule is 00:00 (midnight).

System action: Utility CTOFANCI continues processing.

User response: Determine how the input data was converted. If necessary, adjust the output rule manually.

CTOF46W

M=member D=dsid [L=line] input_parm = "input_value" NOT TRANSLATED

Explanation: The specified input data could not be automatically converted to Control-O parameters by utility CTOFANCI.

The specified data indicates an input parameter or keyword that is not yet supported by utility CTOFANCI. For example, execution of rules on specific dates is controlled in Control-O by calendars. The utility does not generate calendars automatically. If the input script specifies an explicit execution date, the resulting rule does not contain this date reference.

System action: Utility CTOFANCI continues processing.

User response: Determine how the input data was converted. If necessary adjust the output rule manually.

CTOF47W

M=member D=dsid [L=line] input_parm = "input_value" NOT RECOGNIZED

Explanation: The specified input data could not be interpreted by utility CTOFANCI.

This problem can occur for either one of the following conditions:

- The input script contains an error (for example, date specified as 10/10/xx)
- The input rule contains complex logic not supported by utility CTOFANCI.

System action: Utility CTOFANCI continues processing.

User response: Determine how the input data was converted. If necessary adjust the output rule manually.

CTOF90I **PROCESSING COMPLETE *timemsg***

Explanation: This information message indicates that the CTOFANCI utility completed its processing normally.

timemsg is the time when the message was issued.

User response: The time included in the message can be used to help monitor the performance of utility CTOFANCI.

CTOF91W **UNABLE TO VERSION SPACE FROM EMPTY PDS *dsn***

Explanation: This warning message indicates that the CTOFANCI utility was unable to version space from the specified output library during termination.

The specified output library may be empty. An empty output library may exist at the end of CTOFANCI processing if the utility terminated abnormally before writing any members from memory to the output library.

System action: Utility CTOFANCI stops execution.

User response: If necessary, version the library space manually.

CTOF97E **CTOFANCI UNABLE TO COMPLETE TERMINATION PROCESSING – EXITING**

Explanation: After completing or terminating its processing, utility CTOFANCI failed during the cleanup or shutdown operation.

This problem can occur if a file is currently being accessed by another user and the utility is trying to access that file during shutdown.

System action: Utility CTOFANCI stops execution.

User response: To determine the cause of the failure, see other messages in the console log, in report RPTALL, and in the file referenced by DD statement SYSTSPRT. Correct the problem and rerun utility CTOFANCI.

CTOF98E **ERROR WHEN ISSUING MESSAGE**

Explanation: Utility CTOFANCI encountered an error while trying to issue a message.

System action: Utility CTOFANCI stops execution.

User response: To determine the cause of the failure, see other messages on the console log and on the file referenced by DD statement SYSTSPRT. Correct the problem and rerun utility CTOFANCI.

CTOF99E**INTERNAL ERROR *text***

Explanation: Utility CTOFANCI encountered an internal error.

text is a string containing information about the internal error.

System action: Utility CTOFANCI stops execution.

User response: Report the problem to your local BMC Software Customer support representative. Include a copy of the messages from report RPTALL and a copy of the input scripts that were being processed when the failure occurred.

Index

A

- Automation definitions
 - Complexity 29
 - Similar and duplicate 29

B

- BMC Software, contacting 2

C

- Conventions Used in This Guide 10
- Conversion
 - Specific parameters and keywords 34
 - Utility libraries 27
- CTOFANCI
 - Activating utility 43
 - Bypassing problems 69
 - JCL statements 58
 - Monitoring 43
 - Output parameters 45
 - Parameters 44
 - Reports 60
 - Restarting 43
 - Running 26
- customer support 3

I

- INCLUDE Statements 57
 - Sample 57
- Infinite loops
 - Preventing 69
 - Reporting 72

L

- Logic
 - Input 40
 - Output 41

M

- MODE parameter 36

O

- Output dataset names 41
- Output member names 42
- Output parameters
 - Utility CTOFANCI 45
- Output reports 42
- OWNER parameter 37

P

- Parameters
 - Converting specific 34
 - CTOFANCI 44
 - Execution and restart 51
 - MODE 36
 - Output 45
 - OWNER 37
 - Report allocation 49
 - Translation 54
- product support 3

R

- Reports
 - CTOFANCI 60
 - RPTALL 60
 - RPTFILES 61
 - RPTRULES 62
 - RPTSRC
 - Sort-stubs 65
 - Text lines 66
 - Title lines 65
 - RPTWARN 61
- Rules
 - Actions 19
 - Active and Inactive 21
 - Automation
 - Definition 17

Interdependencies 21
Comments 20
Conversion 17
Description 18
Operation Mode 20
Owner ID 20
Scheduling 19
Selection Criteria 18

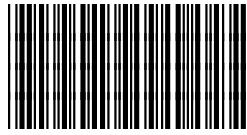
S

Sort-stubs
RPTSRC 66
Statements
INCLUDE 57
TYPERUN 57
support, customer 3

T

technical support 3
TYPERUN Statements 56
Sample 56

Notes



141962