

CA Endeavor[®] Software Change Manager

User Guide

Version 17.0.00



Second Edition

This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

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

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

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

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

The manufacturer of this Documentation is CA.

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

Copyright © 2014 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA Technologies products:

- CA Endeavor® Software Change Manager (CA Endeavor SCM)
- CA Librarian® Base for z/OS (CA Librarian)
- CA Panvalet® for z/OS (CA Panvalet)
- CA Change Manager Enterprise Workbench (CA CMEW)

Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

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

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

Documentation Changes

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

Note: In PDF format, page references identify the first page of the topic in which a change was made. The actual change may appear on a later page.

Version 17.0, Second Edition

- [Printing Element Information Using the Print Elements Action](#) (see page 54)— Updated to add that Print actions that use the Explode option to print input component information extracted from the ACMQ files are processed first before other actions are processed in Type sequence order.

Version 16.0

- [Creating Outputs Using the Generate Elements Action](#) (see page 48)— Updated to include Autogen Span options in the description for Autogen. Added a cross-reference to the scenario "How to Automatically Generate *Using Element*," in the *Scenario Guide*.
- [using elements](#) (see page 136)— Added this term to the Glossary.

Version 15.0

- [View Element Information](#) (see page 76)— Updated to indicate that options to view information about sourceless elements and elements of log or image delta format are limited.
- [View Element Information from a Selection List](#) (see page 77)— Updated to indicate that options to view information about sourceless elements and elements of log or image delta format are limited.

Contents

Chapter 1: Introduction 9

What You Need to Know	9
Audience	9
Typical Software Life Cycle	10
Change Procedures	10
Emergency Operations	11

Chapter 2: Getting Started 13

Start CA Endeavor SCM	13
Exit CA Endeavor SCM	13
Display Online Help	14
Field Default Values.....	14
Selection Lists.....	14
List Options	15
Sort Columns in a Selection List	16
Filter Columns in a Selection List	16
Print a Selection List.....	17
Long Element Names in a Selection List.....	17
Placeholders and Wildcards in a Selection List	18
Dates in a Selection List.....	18
Name Masking.....	19
Wildcards	19
Placeholders.....	19
Valid Uses for Name Masks.....	20
ISPF Standard Commands and Function Keys	21
Command Stacking.....	22
Set User Preferences	23

Chapter 3: Performing Development Tasks in Foreground 25

Development Tasks	25
Execution Report File	26
View the Foreground Execution Report.....	27
Retrieving an Element from an Inventory Location Using the Retrieve Elements Action.....	27
How Retrieving an Element from an Inventory Location Works.....	28
How Retrieving an Element from an Inventory Location Affects the Signout ID	29
How Retrieving an Element from an Inventory Location Affects CCIDs and Comments	29

Retrieve an Element from an Inventory Location	29
Adding a Member from an External Data Set Using the Add Elements Action	30
Considerations When Adding a Member from an External Data Set.....	31
How Adding a Member from an External Data Set Works.....	32
How Adding a Member from an External Data Set Affects the Signout Status	33
How Adding a Member from an External Data Set Affects CCIDs and Comments	34
Add a Member from an External Data Set	34
Updating a Member Using the Update Elements Action	35
How Updating a Member Works	36
Updating a Member and Trailing Blanks in Input Files	37
How Updating a Member Affects the Signout Status	37
How Updating a Member Affects CCIDs and Comments	38
Update a Member.....	38
Moving an Element Between Inventory Locations Using the Move Elements Action	39
Considerations When Moving Elements Using a Generate Processor.....	40
How Moving Elements Between Inventory Locations Works	41
How Moving Elements with History Works	41
How Moving Elements Without History Works	43
How Moving Elements Affects the Signout Status.....	44
How Moving Elements Affects CCIDs and Comments.....	44
Move an Element.....	45
Deleting Element Levels Using the Delete Elements Action	47
How Deleting Element Levels Works	47
Delete Element Levels.....	48
Creating Outputs Using the Generate Elements Action.....	48
Generate an Element Output.....	50
How Generating Elements with Copyback Works	51
How Generating Elements Without Copyback Works	52
How Generating Elements with NoSource Works	52
How Generating Elements Affects the Signout Status.....	53
How Generating Elements Affects CCIDs and Comments.....	54
Printing Element Information Using the Print Elements Action.....	54
Print Element Information	55
Removing the User Signout Using the Signin Elements Action	57
How Removing the User Signout Works	57
How Removing the User Signout Affects CCIDs and Comments.....	57
How Removing the User Signout Affects the Signout ID	58
Remove the User Signout.....	58
Chapter 4: Packaging Element Actions	61
Packages	61

Create a Package	61
Cast a Package	62
Review and Approve a Package	63
Execute a Package	64
Submit a Package for Batch Processing	66

Chapter 5: Performing Batch Tasks in Foreground **67**

Perform Batch Tasks in Foreground	67
Build Element Action SCL in Foreground for Execution in Batch	67
Edit Batch Requests	70
Validate Batch Jobs	71
Build Additional JCL	71
Submit Batch Job for Processing	72

Chapter 6: Viewing Inventory Information **75**

Inventory Components	75
View Element Information	76
View Element Information from a Selection List	77
View Footprint Information	81
View Site Definitions	82
View Stage Information	82
View System Definitions	83
View Subsystem Definitions	83
View Type Definitions	84
View Processor Group Definitions	85
View Approver Group Definitions	86
View Approver Group Relationship Definitions	87
View Environment Information	88
View Site Symbolics	88
View Global Type Processing Sequence	89

Appendix A: Selecting the Correct Processor Group **91**

Element Actions and Processor Groups	91
Processor Group Information	93
Variables in Processor Group Information	94
Processor Group Rules	95
Processor Group Example Criteria	95
Add Elements Action and Processor Groups	96
Add Elements Action, Case 1 Example	96
Add/Update Elements Actions and Processor Groups	97

Add/Update Elements Action, Case 1 Example	98
Add/Update Elements Action, Case 3 Example	98
Delete Elements Action and Processor Groups.....	98
Delete Elements Action, Case 1 Example	99
Generate Elements Action and Processor Groups	99
Generate Elements Action, Case 1 Example	100
Generate Elements Action, Case 4 Example	100
Generate Elements with Copyback and Processor Groups	101
Generate Elements with Copyback Action, Case 1 Example	101
Restore Elements Action and Processor Groups.....	102
Restore Elements Action, Case 1 Example	102
Move Elements Action (Using a Move Processor) and Processor Groups	103
Move Elements Action (Using a Move Processor), Case 1 Example	105
Move Elements Action (Using a Move Processor), Case 7 Example	105
Move Elements Action (Using a Generate Processor) and Processor Groups	105
Move Elements Action (Using a Generate Processor), Case 3 Example	107
Move Elements Action (Using a Generate Processor), Case 5 Example	107
Transfer Elements Action (Using a Move Processor) and Processor Groups	108
Transfer Elements Action (Using a Move Processor), Case 1 (A1) Example	110
Transfer Elements Action (Using a Move Processor), Case 7 Example	110
Transfer Action (Using a Generate Processor) and Processor Groups.....	111
Transfer Elements Action (Using a Generate Processor), Case 1 (A2) Example.....	113
Transfer Elements Action (Using a Generate Processor), Case 8 Example	114

Glossary	115
-----------------	------------

Index	139
--------------	------------

Chapter 1: Introduction

This section contains the following topics:

[What You Need to Know](#) (see page 9)

[Audience](#) (see page 9)

[Typical Software Life Cycle](#) (see page 10)

[Change Procedures](#) (see page 10)

[Emergency Operations](#) (see page 11)

What You Need to Know

CA Endeavor SCM is an integrated set of management tools used to automate, control, and monitor the mainframe software development life cycle.

CA Endeavor SCM runs under z/OS, within the CICS or TSO/ISPF environments and in batch. Your site can also run an API program that interfaces with CA Endeavor SCM.

To use the product, you need a working knowledge of the mainframe environment, the z/OS mainframe operating system, Time Sharing Option facility (TSO), and the Interactive System Productivity Facility (ISPF). It is also assumed that the product has been properly installed and configured at your site.

This guide describes the tasks that users can perform in the ISPF environment. Foreground and many batch tasks can be performed using the ISPF panel driven interface. Batch JCL streams can also be written using the standard ISPF editor and submitted from TSO/ISPF.

Note: For more information about writing batch JCL streams, see the *SCL Reference Guide* and the *Packages Guide*.

Note: For more information about terms used in this guide, see the "Glossary."

Note: For information about installing and configuring the product, see the *Installation Guide*.

Audience

Application developers, quality control engineers, and managers (development managers, quality control managers, product and project managers, and production control managers) benefit from using the product in the following ways:

- Application Developers use element actions to develop, manage, and control software changes. For example, based on an approved change order, a developer uses the Retrieve Elements action to retrieve an element from the production stage to a development library. After changing the source code, the developer uses the Add Elements action to add the element to the testing stage. Finally, the developer can use the Generate Elements action to create an executable form of an element.
- Development, Quality Control, and Source Control Managers use element actions to approve and move elements through the stages in the software life cycle. For example, after all functional testing has been performed on source code, a Quality Control Manager can create a package to promote all of the code comprising a specific piece of software functionality to the production stage.

Typical Software Life Cycle

You can automate and control the movement of software through your software life cycle. Software life cycles are site-specific. A typical software life cycle consists of the following stages:

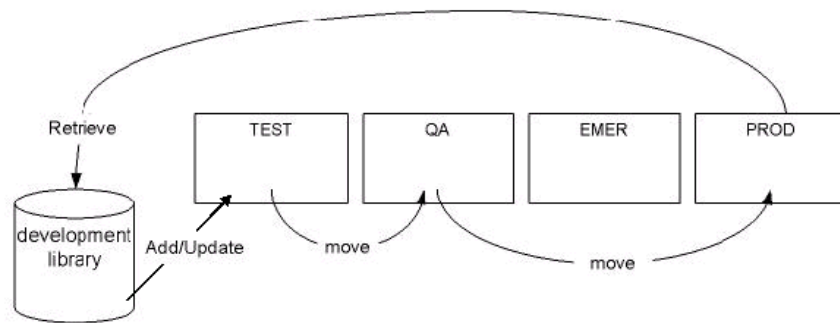
- Development. Applications are developed in this library.
- Test. Applications are unit tested in this stage.
- Quality Assurance. Applications are system tested in this stage.
- Emergency. Fixes are applied to production code in this stage.
- Production. Production applications are stored in this stage.

Change Procedures

The typical change procedures that the product manages involve the following activities:

- Retrieving elements from production to a development library
- Making changes to elements
- Adding or updating elements into the test stage
- Moving elements to QA
- Moving elements back into production

The following illustration shows the typical change procedures in a software life cycle.



In this life cycle, new or existing code is developed in the Development library. After the code has been unit tested by the application developer, the code is added to the Test stage for unit testing. When the code is ready for system testing, it is moved to the Quality Assurance stage. Finally, when the code is ready to be used by customers, the code is moved to the Production stage. Any fixes are applied to production code in the Emergency stage.

If your company requires approvals to move your software code changes through the life cycle, you must use packages to perform the moves.

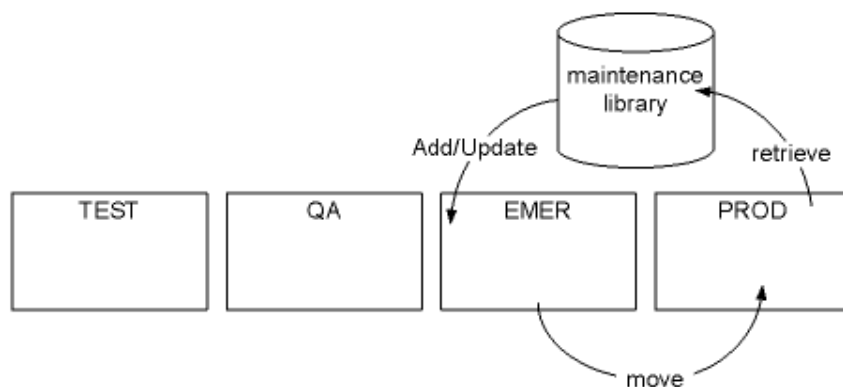
Note: For more information about packages, see the *Packages Guide*.

Emergency Operations

The product can manage the following emergency operations when fixes must be applied to production code:

- Retrieving elements from production
- Making changes to elements
- Adding or updating elements into the emergency stage
- Moving elements to production

The following illustration shows emergency change procedures in a software life cycle.



In this life cycle, production code is retrieved from the Production stage and is added back to the Emergency stage. After the fixes are applied to the code in the Emergency stage, the code is moved back to the Production stage.

If your company requires approvals to move your software code changes through the life cycle, you must use packages to perform the moves.

Note: For more information about packages, see the *Packages Guide*.

Chapter 2: Getting Started

This section contains the following topics:

[Start CA Endeavor SCM](#) (see page 13)

[Exit CA Endeavor SCM](#) (see page 13)

[Display Online Help](#) (see page 14)

[Field Default Values](#) (see page 14)

[Selection Lists](#) (see page 14)

[Name Masking](#) (see page 19)

[ISPF Standard Commands and Function Keys](#) (see page 21)

[Command Stacking](#) (see page 22)

[Set User Preferences](#) (see page 23)

Start CA Endeavor SCM

To access the product, follow the instructions provided by your Endeavor administrator, because access depends on your site's specific implementation.

Exit CA Endeavor SCM

You can exit the product at any time by entering **X** in the Option field on the Primary Options panel.

Note: If the INTERCEPT ISPF RETURN CMD option on the User Defaults panel is set to Y, use RETURN or JUMP from any panel to exit the product immediately. If the INTERCEPT ISPF RETURN CMD option on the User Defaults panel is set to N or Y, use RETURN or JUMP from the Primary Options panel to exit the product immediately.

Note: For more information about setting user defaults, see Set User Preferences.

Display Online Help

You can display online help to read more information about messages, features, panel options and fields, or ISPF fields. Use the following methods to display online help:

To display expanded messages

1. Press PF1, when abbreviated message text is displayed on a panel. The expanded message provides more information than the abbreviated message.
2. (Optional) Press PF1. The help text for the current panel opens.

To display tutorials about the product

Enter **T** in the Option field on the Primary Options panel and press Enter. Tutorials about product features are displayed.

To display information about panel options or fields, or ISPF fields

Press PF1 from any panel. The help text for that panel opens. The help page displays information about panel options and fields, or ISPF fields.

Field Default Values

Selection Lists

When you enter the information on a panel to process a request, if you leave the system, subsystem, or element fields blank, or use a *name mask* on an action panel (and the DISPLAY LIST option is set to Y), a System, Subsystem, Element, or Member Selection List displays. You can use these lists to fully qualify your request.

After you select the elements or members for which you want to process your request, your requested element action is executed.

Element Selection lists are sorted alphabetically by element name, stage, system, subsystem, and type. Whenever possible, you should fully qualify a system when requesting the selection list. If you cannot fully qualify a system, carefully examine the location (environment and stage) and classification information (system, subsystem, and type) associated with an element to make sure it is the element you want.

Note: If you enter all of the information for your request, except for the stage, select an option, and the result has only one element, the Element Selection List does not display.

More information:

[Filter Columns in a Selection List](#) (see page 16)

[Print a Selection List](#) (see page 17)

[Dates in a Selection List](#) (see page 18)

[Long Element Names in a Selection List](#) (see page 17)

[Placeholders and Wildcards in a Selection List](#) (see page 18)

List Options

When you enter information on various display and element action panels, list options are available. If you select the list option Apply List Filters, then the Apply List Filters panel opens. On this panel, you can filter the list by CCID, user ID, and by processor group name. For example, you can use the user ID filter to find elements previously acted upon (modified, generated, or retrieved) by you, or by someone else, depending on the user ID you specify. You can further limit the list by eliminating CCID and user ID matches for any of the following categories: current, generate, last action, or retrieve.

Sort Columns in a Selection List

To make it easier to view a selection list, you can sort the rows by the content of a particular column. The ESORT command lets you specify a column and whether the content of that column is to be sorted in ascending or descending order.

Note: The ESORT function has to be enabled by your Endeavor administrator.

To sort a selection list, use one of the following methods.

To sort a selection list in ascending order

Enter **ES** in the Command field, followed by the first three letters of the column heading, and then press Enter. The rows of the selection list are sorted in ascending order based on the content of the column you specified.

For example, to sort a selection list by environment in ascending order, enter ES ENV

To sort a selection list in descending order

Enter **ES** in the Command field, followed by the first three letters of the column heading preceded by a minus sign (–), and then press Enter. The rows of the selection list are sorted in descending order based on the content of the column you specified.

For example, to sort a selection list by system in descending order, enter ES –SYS

To sort a selection list by the default sort order

Enter **ES** in the Command field and press Enter. The rows of the selection list are sorted in ascending order based on the content of the default column, which is usually the first column.

Note: You can also enter **TSO ESORT**.

Filter Columns in a Selection List

You can use the EONLY command to filter a selection list by including or excluding rows that match the column filter value that you specify. Before using EONLY, you must build a selection list for a scrollable list panel.

When you enter the column name, you only have to enter the first three characters (with a maximum of six characters). In addition, the value field supports placeholders. Finally, you can filter columns using EONLY or EO (as long as the supplied CTLICMDS table is in use) and using TSO EONLY.

Important! After you filter a selection list, the only way to remove the filter is to press PF3. You must then rebuild the list.

To filter a selection list

The following table lists the different ways you can use **EONLY** to filter a selection list:

To filter	Do the following
All rows including a specific column value	Enter EONLY <i>columnname value</i> in the Command field, and press Enter. For example, enter EONLY TYPE CBL to filter an Element Selection List to display only those elements of type CBL (for COBOL code).
All rows that do not include a specific column value	Enter EONLY - <i>columnname value</i> in the Command field, and press Enter. For example, enter EONLY - TYPE ASMPGM to filter an Element Selection List to display all elements except those of type ASMPGM (for Assembler code).

Print a Selection List

You can print a selection list at any time by entering **EPRINT** in the Command field, which sends the output to your ISPF listing data set.

Note: You can also print a selection list by entering **TSO EPRINT** in the Command field.

Long Element Names in a Selection List

You cannot work with long element names from an ISPF selection list. Access to long element names is only available using CA CM Enterprise Workbench.

When long element names appear in a CA Endeavor SCM ISPF selection list, element names that are longer than ten characters are truncated, and only the first five characters of the element name appear in the list. If you have more than one long element name that begins with the same five characters, the ISPF selection list displays only one element.

Placeholders and Wildcards in a Selection List

Before you use placeholders [the percent (%) sign] and wildcard characters [the asterisk (*) character] in a selection list, consider the following information to ensure your success:

- When filtering columns in a selection list, you can use EONLY with the percent sign (%) placeholder character to represent any single character in a string. In addition, you can use the percent sign as a placeholder multiple times in a single field, for example, C%CC%C. However, any value that you enter in the value field is treated as if it ends in the wildcard character. Therefore, you cannot use EONLY to filter rows containing a specific number of characters less than the maximum of six characters. For example, the values C, C%, and C%%% will return the same result.
- When filtering columns in a selection list, you cannot use EONLY with the asterisk (*) wildcard character because the asterisk is treated as a character. For example, the value C* will produce an empty list, unless there is an item in the list that contains an asterisk in the second position.

Dates in a Selection List

When you use dates and date ranges in a selection list, you specify the dates in the value field of EONLY in the format DDMMYY (for example, 30SEP01). In addition, you can filter all rows that have dates earlier than or later than the date in the value field, as well as the rows that are between two dates specified in the value field.

Example: Filter all rows with a date later than December 31, 2001

This example shows how to use the greater than sign (>) to filter all rows containing a date later than December 31, 2001.

```
EO DAT >31DEC01
```

Example: Filter all rows with a date earlier than January 1, 2002

This example shows how to use the less than sign (<) to filter all rows containing a date earlier than January 1, 2002.

```
EO DAT <01JAN02
```

Example: Filter all rows containing a date between January 1 and June 30, 2001

This example shows how to use a dash (-) to filter all rows containing a date between January 1 and June 30, 2001.

```
EO DAT 01JAN01-30JUN01
```

Name Masking

To help you more easily find information and process requests, you can use *name masking*. By substituting a name with the asterisk wildcard character (*), a character with the percent sign placeholder (%), or by using both together, it is much easier to find information and process requests.

Wildcards

A wildcard is an asterisk (*) character that can be used in a search string to represent the entire search string or the end of a search string. It represents any number of characters. When a wildcard is used as the only character of a search string, all members of the search field are returned. When a wildcard is used as the last character of the search string, the only members of the search field returned are those that begin with the characters in the search string preceding the wildcard.

You cannot have more than one wildcard in a string. For example, the statement `ADD ELEMENT U*PD*` would result in an error.

Examples: Use a Wildcard

- This example shows how to use a wildcard as the only character in a search string. This command adds all elements.
`ADD ELEMENT *`
- This example shows how to use a wildcard as the last character of a search string. This command adds all elements beginning with UPD such as UPDATED or UPDATE.
`ADD ELEMENT UPD*`
- This example shows how to use the asterisk wildcard as the last character of a search string to select all package IDs beginning with PKG, such as PKGS, PKGB, PKGC, PKGB2, and PKGC5A.
`PKG*`

Placeholders

A placeholder is a percent sign (%) character that can be used to represent one character in a search string. It can be used at the end of a search string, multiple times within a search string, or both. When a placeholder is used as the last character in a string, all members of the search field are returned, beginning with the characters in the search string preceding the placeholder, but which have no more characters than were used in the search string.

Examples: Use a Placeholder

- This example shows how to use a placeholder to add all elements with four-character names beginning with UPD such as UPD1 or UPDA.

ADD ELEMENT UPD%

- This example shows how to use the percent sign placeholder to return all package IDs beginning with PKG, such as PKGS, PKGB, and PKGC.

PKG%

- This example shows how to use the percent sign placeholder multiple times in a single search string to return all elements with five-character names that have the letter U as the first character and PD as the third and fourth character.

ADD ELEMENT U%PD%

Examples: Use a Wildcard and a Placeholder Together

- This example shows how to use both the wildcard and placeholder to add elements with names of any length that have U as the first character, any one character as the second character, and D as the third character.

ADD ELEMENT U%D*

- This example shows how to use both the wildcard and placeholder to select all package IDs that have P as the first character, any one character as the second character, and G as the third character, for example, PKGABCD, POGS, PIGGY, PPG1234NDVR, and so on.

P%G*

Valid Uses for Name Masks

You can use name masks for the following items:

- On API list requests, name masking is valid on inventory locations (environment, system, subsystem, type name, and stage ID)
- On CSV utility SCL requests, name masking is valid on inventory locations (environment, system, subsystem, type name, and stage ID)
- Element names. However, element name masks are **not** valid under the following conditions:
 - When entering a LEVel in a statement
 - When using the MEMber clause with an action statement
 - When building a package

- Environment name masks are valid on the following:
 - On API list requests
 - On CSV utility SCL requests
 - On requests for historical reports
 - On SCL statements, only for the FROM ENVIRONMENT field of the WHERE COMPONENTS EQUAL clause
 - On ISPF panels when listing elements, except that environment name masks are **not** valid under the following circumstances:
 - On Add/Update panels
 - On Generate panels, the option COPYBACK cannot be specified when the Environment name is masked
 - On any panels where environment name masking is allowed, the following list options are ignored when you mask the environment name: BUILD USING MAP, RETURN FIRST FOUND, DISPLAY SYS/SBS LIST
- ISPF panels
- System, subsystem, and type names within FROM clauses
- Report syntax, except environment name masking is only allowed on historical reports
- Package IDs
- Parallel Development Option (PDM) does support name masks on CA Endeavor SCM inventory locations (environment, system, subsystem, type, and stage)

ISPF Standard Commands and Function Keys

There are several standard ISPF commands and function keys that can help you more easily move through the product. The following describes the commands and function keys that are available from any ISPF panel.

Note: For more information about these commands and key assignments, see your ISPF documentation.

ENTER (command), ENTER (function key)

Processes the information on the panel.

REFRESH (command), PA2 (function key)

Refreshes the screen.

HELP (command), PF1 (function key)

Displays help and expands the abbreviated help message.

END (command), PF3 (function key)

Returns to the previous logical panel. From the Primary Options panel (highest level), returns to ISPF or TSO, as appropriate. To cancel the request and return to the previous panel, press END.

Note: The END command (PF3) is not affected by the INTERCEPT ISPF RETURN CMD field on the User Defaults panel. For example, the Environment Selection panel appears when you use the END command from the Primary Options panel.

RETURN (command), PF4 (function key)

Returns to the ISPF Primary Options panel.

Note: If you use RETURN or JUMP (=) commands from the Primary Options panel, the ISPF Primary Option panel appears. When you set INTERCEPT ISPF RETURN CMD =Y on the User Defaults panel and you use RETURN or JUMP commands from any panel other than the Primary Options panel, the Primary Options panel appears. Finally, if you use the JUMP command from within the product when INTERCEPT ISPF RETURN CMD = Y, any command you use after the equal sign (=) will be ignored by ISPF. This means that if you use JUMP from any panel other than the Primary Options panel, the Primary Options panel appears.

UP (command), PF7 (function key)

Scrolls up one page to see lines that are not visible at the top of the panel.

DOWN (command), PF8 (function key)

Scrolls down one page to see lines that are not visible at the bottom of the panel.

LEFT (command), PF10 (function key)

Scrolls left.

RIGHT (command), PF11 (function key)

Scrolls right.

Command Stacking

To help improve your navigation speed through the ISPF panels, you can stack commands together (known as *command stacking*) by entering multiple commands separated by a semicolon, rather than by entering single commands. For example, if you are using the Type Definition panel and you want to navigate to the Add/Update Elements panel, you must press the End key until you reach the Primary Options panel, select the FOREGROUND option, and then select the ADD/UPDATE option. However, if you stack the commands together using command stacking, it is much faster to navigate to the panel you want to use.

This command has the following format:

```
[command];[option]
```

command

(Optional) Specifies the command for navigation. Separate multiple commands with a semicolon.

option

(Optional) Specifies the option for navigation. Separate multiple options with a semicolon.

Example: Navigate to the Add/Update Elements Panel

This example shows how to navigate from the Type Definition panel to the Add/Update Elements panel.

```
END;END;2;2
```

Note: For more information about command stacking, see your ISPF documentation.

Set User Preferences

You can set user preferences to allocate space for work and list data sets, define printer output class and page size settings, record jobcard settings when submitting batch print jobs, set a threshold for the Execution Report, specify how ISPF RETURN and JUMP commands should work, and specify whether to use the ISPF browse or view display services.

Note: If you use the JUMP command from the Primary Options panel, you will jump to the application, regardless of how you have set the INTERCEPT ISPF RETURN CMD option.

To set user preferences

1. Start the product using the instructions provided by your site administrator.
The Primary Options panel appears.
2. Enter **0 (Defaults)** and press Enter.
The User Defaults panel appears.

3. Select your preferences.

Note: For more information about a field or option on the panel, press PF1.

4. Press Enter.

Your preferences are saved, and the product will work this way until you change the values in this session, or in a subsequent session.

Chapter 3: Performing Development Tasks in Foreground

This section contains the following topics:

[Development Tasks](#) (see page 25)

[Retrieving an Element from an Inventory Location Using the Retrieve Elements Action](#) (see page 27)

[Adding a Member from an External Data Set Using the Add Elements Action](#) (see page 30)

[Updating a Member Using the Update Elements Action](#) (see page 35)

[Moving an Element Between Inventory Locations Using the Move Elements Action](#) (see page 39)

[Deleting Element Levels Using the Delete Elements Action](#) (see page 47)

[Creating Outputs Using the Generate Elements Action](#) (see page 48)

[Printing Element Information Using the Print Elements Action](#) (see page 54)

[Removing the User Signout Using the Signin Elements Action](#) (see page 57)

Development Tasks

When you need to perform a development task in CA Endeavor SCM, you perform an action on an *element*. For example, to manage a member from an external data set, you use the Add Elements action. The process that you use to perform a development task is similar for each element action. You specify the action (for example, add an element, move an element), enter information associated with the task you want to perform (for example, specify the library, data set name, element name, system, and subsystem), and then execute the action.

For each development task you perform, CA Endeavor SCM performs the following actions:

- *Source management*, which relates to the maintenance of the element source, or the updating of base and delta libraries.
- *Inventory management*, which relates to the maintenance of Master Control File (MCF) definitions.
- *Output management*, which relates to the creation and maintenance of outputs, including object modules, load modules, listings, and so forth.

Note: When you perform a development task and enter information (for example, the Environment, System, Subsystem, Element, Type, and Stage fields), make sure that you enter names using the national character set (A-Z, 0-9, @, #, and \$) and consider using a *name mask*. For more information about name masking, see [Name Masking](#) (see page 19).

Some element actions are available in foreground, in batch, and in packages. This section contains information for performing development tasks in the *foreground* only.

Note: For more information about how to perform an element action in batch, see the *SCL Reference Guide*. For more information about how to perform an element action using packages, see the *Packages Guide*. The Add, Move, Generate, and Signin actions can be performed from the Quick-Edit Option panel, if the CA Endeavor SCM Quick Edit option is enabled at your site. For more information about Quick-Edit, see the Quick Edit Option User Guide.

Execution Report File

After you execute a foreground action request, an Execution Report is generated and you may see a message that specifies the file name of the Execution Report. This message appears in the following format:

Browse *userid.C1TEMPRn.Esysid.MSGS*.

userid.C1TEMPRn.Esysid.MSGS

Indicates the name of the file containing the Execution Report for the requested action.

userid

Indicates your ISPF user ID.

n

Indicates a system-assigned number between one and nine.

sysid

Indicates the TSO system on which the element action is being processed.

View the Foreground Execution Report

The Execution Report is displayed whenever the return code from foreground action processing is greater than, or equal to, a threshold value.

If the report is not automatically displayed because the threshold value was not met, you can still view the report. To view the Execution Report, type **QMSGGS** (or **QM**) in the command field or use the ISPF Browse facility. The following file contains the Execution Report for the requested action:

Browse *userid.C1TEMPRn.Qsysid.MSGS*

userid – Specifies your ISPF user ID.

n – Specifies a system-assigned number between 1 and 9.

sysid – Specifies the SYSID of the LPAR where you are using CA Endeavor SCM.

Note: You can change the threshold value that determines when the Execution Report is automatically displayed. For more information, see Set Dialog Defaults.

More information:

[Set User Preferences](#) (see page 23)

Retrieving an Element from an Inventory Location Using the Retrieve Elements Action

To retrieve an element from an inventory location to an external (user) data set, you use the Retrieve Elements action (available in foreground, batch, and in packages).

This section contains information for performing this development task in the *foreground* only.

Note: For more information about how to perform this element action in batch, see the *SCL Reference Guide*. For more information about how to perform this element action using packages, see the *Packages Guide*. The Add, Move, Generate, and Signin actions can be performed from the Quick-Edit Option panel, if the CA Endeavor SCM Quick Edit option is enabled at your site. For more information about Quick-Edit, see the *Quick Edit Option User Guide*.

More information:

[How Retrieving an Element from an Inventory Location Affects the Signout ID](#) (see page 29)

[How Retrieving an Element from an Inventory Location Affects CCIDs and Comments](#) (see page 29)

[Retrieve an Element from an Inventory Location](#) (see page 29)

How Retrieving an Element from an Inventory Location Works

When you retrieve an element from an inventory location to an external (user) data set using the Retrieve Elements action, the product performs the following actions and causes these effects:

1. Determines whether the element is signed out. If the element is signed out to someone else, and if you do not set the SIGNOUT ELEMENT option to N, you must set the OVERRIDE SIGNOUT option to Y to retrieve the element.

Note: Your site administrator must set up permissions for you to use the OVERRIDE SIGNOUT option.

2. Searches for the specified element, in the following order:

- At the location you specify in the RETRIEVE elements action
- In each stage in the environments on the map route

Note: If the product finds the element at a subsequent stage that is not part of the map, you get a warning message. In addition, if you do not know the exact location of the element you want to retrieve, specify the lowest stage in the map route. The product searches the entire map route for the element.

3. Copies the current level of the element to the output data set. If you set the EXPAND INCLUDES option to Y, any INCLUDE statement is expanded when it is copied from the source. If the output data set is a library, the following applies:
 - If the member currently exists in the library, you must set the REPLACE MEMBER option to Y. Otherwise, the RETRIEVE action fails.
 - By default, the assigned member name will be the same as the element name you enter. You can enter a different name for the member. In this situation, the product assigns the new name to the retrieved element in the user library.
4. If you set the SIGNOUT ELEMENT option to Y, the product updates the Master Control File (MCF) and signs the element out to you. If you set the SIGNOUT ELEMENT option to N, the product stops processing the action after it creates the copy in step 3.

How Retrieving an Element from an Inventory Location Affects the Signout ID

When you retrieve an element from an inventory location to an external (user) data set using the Retrieve Elements action, the software signs the element out to you by recording your user ID in the SIGNOUT ID field on the Master Control File (MCF) for the stage from which the element was retrieved, provided you set the SIGNOUT ELEMENT option to Y. You should only set the SIGNOUT ELEMENT option to N when you want a copy of the element to review, but do not want to make changes to the element.

The administrator activates the signin/signout option for each system. If signin/signout is not in effect, then the signout user ID is always updated.

How Retrieving an Element from an Inventory Location Affects CCIDs and Comments

When you retrieve an element from an inventory location to an external (user) data set using the Retrieve Elements action, you can enter a CCID and comment. When you enter a CCID and comment, the Retrieve CCID/Comment is set. When you set the SIGNOUT ELEMENT option to Y, the product records the CCID and comment you entered in the RETRIEVE CCID and COMMENT fields in the Master Control File (MCF). If you set the SIGNOUT ELEMENT option to N, the product does not update this information.

Retrieve an Element from an Inventory Location

To retrieve an element from an inventory location to an external (user) data set, you use the Retrieve Elements action.

Important! If your site administrator has enabled Global Type Sequencing, element actions execute in the type sequence order defined by your administrator (at the site level), regardless of the action's inventory location.

To retrieve an element from an inventory location

1. Start the product using the instructions provided by your site administrator.
The Primary Options panel appears.
2. Enter **2** (Foreground) and press Enter.
The Foreground Options menu appears.
3. Enter **3** (Retrieve) and press Enter.
The Retrieve Elements panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. Choose *one* of the following steps.
 - a. To retrieve an element, enter the information for the action, enter **R** in the Option field, and press Enter.
 - b. To retrieve an element from an Element Selection List (applies when you set the DISPLAY LIST option to Y), enter the information for the action, leave the Option field blank, and press Enter.
 - c. To retrieve a prior version of an element, enter the information for the action, enter **S** in the Option field, and press Enter.
5. (Optional) If the Element Selection List or Summary of Levels panel appears, enter an option to the left of each member you want to retrieve and press Enter. The ISPF Locate command is supported on Element Selection lists.

The element is retrieved to an external (user) data set.

Adding a Member from an External Data Set Using the Add Elements Action

To control and manage a member from an external data set, you use the Add Elements action (available in foreground, batch, and in packages). The member can be any of the following:

- A sequential data set
- Hierarchical File System (HFS) files (CA CM Enterprise Workbench only)
- PDS or PDSE library members
- CA Panvalet library members
- CA Librarian library members
- Load modules (RECFM=U)
- Binary files

You can add members only to the entry stage, and only when there is no other element with the same name in the target entry stage.

Note: If you have an element with the same name in the target entry stage, use the Update Elements action.

This section contains information for performing this development task in the *foreground* only.

Note: For more information about how to perform this element action in batch, see the *SCL Reference Guide*. For more information about how to perform this element action using packages, see the *Packages Guide*. The Add, Move, Generate, and Signin actions can be performed from the Quick-Edit Option panel, if the CA Endeavor SCM Quick Edit option is enabled at your site. For more information about Quick-Edit, see the *Quick Edit Option User Guide*.

More information:

[How Adding a Member from an External Data Set Works](#) (see page 32)

[How Adding a Member from an External Data Set Affects the Signout Status](#) (see page 33)

[How Adding a Member from an External Data Set Affects CCIDs and Comments](#) (see page 34)

[Add a Member from an External Data Set](#) (see page 34)

Considerations When Adding a Member from an External Data Set

Before you allow the product to control a member from an external data set using the Add Elements action, consider the following information to ensure your success:

Important! If you do not follow these considerations and continue adding the element, the Add Elements action will fail.

- Ensure that an element with the same name is not already in the target entry stage. If so, first contact the current element owner, or your site administrator, to discuss the most appropriate action. Then, consider these workarounds:
 - Wait until the element has been moved out of the entry stage, and then add the member.
 - Add the member using the UPDATE IF PRESENT option.
 - Add the member using the Update Elements action.

- Ensure that the element corresponding to the member you want to add is signed out to you. If the element is signed out to another person on your team, first contact the current element owner, or your site administrator, to see if they can sign the element out to you. Then, consider these workarounds:
 - Sign the element in using the SIGNOUT TO option.
 - Add the member using the OVERRIDE SIGNOUT option.
- Note:** If your site administrator has restricted permissions for the OVERRIDE SIGNOUT option, you may not be able to use the option.

How Adding a Member from an External Data Set Works

When you allow the product to control a member from an external data set using the Add Elements action, the product performs the following actions and causes these effects:

1. Verifies that the element is signed out to you and does not exist in the entry stage.
2. Searches beyond the entry stage for an element with the same name.

Note: Every stage in the mapped route is searched for a match. If the element is found in a stage in the map, the product copies the current version of the element to the target entry stage, along with the last processor group name used for the element.

3. If you code the new version option, the element is *not* copied back to the entry stage. The product assigns the version number you enter.

Important! If the element is found in a stage that is *not* included in the map, you get a warning message. The search continues for an element in a stage that is included in the map.

4. Compares the member being added with the entry stage base. If the element is not found along the map, the element is created in the entry stage with a version number of 01 and base level of 00 (zero).
5. If the product built an entry stage base, the member being added is compared to that base, and builds a new level with any changes. If no changes are detected, you get a warning message.
6. Updates the Master Control File (MCF).

Important! The value specified for Signout Upon Fetch (the SOFETCH parameter) in the Defaults Table (C1DEFULTS) effects how the MCF will be updated for the element copied back (fetched). If Signout Upon Fetch is in effect, the element will be signed out to you unless it is already signed out to someone else. If Signout Upon Fetch is not in effect, the element will not be signed out to you.

Note: For more information about Signout Upon Fetch, see the *Administration Guide*.

7. Continues executing the action based on the value in the GENERATE ELEMENT option. If this option is set to N, the element is *not* generated. If this option is set to Y, the product performs the following actions:
 - Reads the type definition for a source output library specification and writes a copy of the current level of the element to that library. If the EXPAND INCLUDES option is set to Y, the product expands INCLUDE statements in the source.
 - Determines which processor group to use and executes the generate processor in that group (if one has been specified). After the generate processor has been run for the element, the product updates the processor information in the Master Control File.

Note: For more information about processors, see the appendix Selecting the Correct Processor Group.
8. Deletes the member in the FROM library (if the DELETE INPUT SOURCE option is set to Y).

How Adding a Member from an External Data Set Affects the Signout Status

Your site administrator can enable a signout capability on a system-by-system basis. When you allow the product to control a member from an external data set using the Add Elements action, the product performs the following actions and causes these effects:

- If the element exists up the map, a *fetch* is performed.
 - If your site administrator has enabled Signout Upon Fetch, the fetched (source) element is signed out to you, if the element is not signed out to someone else. In addition, the target element is signed out to you.
 - If your site administrator has not enabled Signout Upon Fetch, there is no change in the signout status, and the target element is signed out to you.
- If the element does *not* exist up the map, a fetch is not performed.
 - If your site administrator has enabled Signout Upon Fetch, the target element is signed out to you.
 - If your site administrator has not enabled Signout Upon Fetch, the target element is signed out to you.

Note: For more information about Signout Upon Fetch, see the *Administration Guide*.

How Adding a Member from an External Data Set Affects CCIDs and Comments

When you allow the product to control a member from an external data set using the Add Elements action, you can enter a CCID and comment. When you enter a CCID and comment, the product sets the following options:

If you manage a new element:

- Current Source CCID/Comment
- Generate CCID/Comment (if changed)
- Last Action CCID/Comment
- Source Delta CCID/Comment
- Component Delta CCID/Comment (if generated)

If you manage an existing element:

- Current Source CCID/Comment (if changed)
- Generate CCID/Comment (if generated)
- Last Action CCID/Comment
- Source Delta CCID/Comment
- Component Delta CCID/Comment (if the generate creates a delta)

Note: If you set the GENERATE ELEMENT option to N, the Add Elements action does not set the generate or component delta CCIDs and comments.

Add a Member from an External Data Set

To control and manage a member from an external data set (PDS or PDS/E member, CA Panvalet, CA Librarian or sequential data set, or load module), you use the Add Elements action.

Important! When you enter information for the Add Elements action, if you enter a member name that already exists, the action is rejected or the previous member is replaced with the one you are currently adding (based on the value in the UPDATE IF PRESENT option).

To add a member from an external data set

1. Start the product using the instructions provided by your site administrator.
The Primary Options panel appears.
2. Enter **2** (Foreground) and press Enter.
The Foreground Options menu appears.

3. Enter **2** (Add/Update) and press Enter.

The Add/Update Elements panel appears.

Note: For more information about a field or option on the panel, press PF1.

Note: For more information about adding or updating an element with the Autogen option, see How Generating Elements with Autogen Works.

4. Choose *one* of the following steps.
 - a. To add a member, enter the information for the action, enter **A** in the Option field, and press Enter.

Note: If you want the element name to be different from the member name, change the name in the ELEMENT option. In addition, if you are adding an element for the first time, the comment becomes the element description in the Master Control File.

- b. To add multiple members, leave the Option field blank and press Enter.

The Member Selection List appears. This list appears when the source data set is a library, you leave the Option field blank, the DISPLAY LIST option is set to **Y**, and when both the ELEMENT and MEMBER options are blank or have a name mask. The ISPF Locate command is supported on Member Selection lists.

5. (Optional) If the Member Selection List appears, enter **A** to the left of each member you want to add and press Enter.

The members are added and you return to the Add/Update Elements panel. You can browse the listing data set for more information about each selected member.

Updating a Member Using the Update Elements Action

To control and manage a member when an element with the same name already exists in the target entry stage, you use the Update Elements action (available in foreground, batch, and in packages).

This section contains information for performing this development task in the *foreground* only.

Note: For more information about how to perform this element action in batch, see the *SCL Reference Guide*. For more information about how to perform this element action using packages, see the *Packages Guide*. The Add, Move, Generate, and Signin actions can be performed from the Quick-Edit Option panel, if the CA Endeavor SCM Quick Edit option is enabled at your site. For more information about Quick-Edit, see the *Quick Edit Option User Guide*.

More information:

[Updating a Member and Trailing Blanks in Input Files](#) (see page 37)

[How Updating a Member Affects the Signout Status](#) (see page 37)

[How Updating a Member Affects CCIDs and Comments](#) (see page 38)

[Update a Member](#) (see page 38)

How Updating a Member Works

When you allow the product to control a member when an element with the same name already exists in the target entry stage using the Update Elements action, the product performs the following actions and causes these effects:

- Verifies that the element currently exists in the entry stage. If the element does not exist in the entry stage, the update fails.
- Determines whether the element is signed out to you. If the element is signed out to someone else, you must set the OVERRIDE SIGNOUT option to Y to update the element.

Note: If the element is in the entry stage, but it is identical to the update source element, you get a warning message saying "NO CHANGES," and no level is created.

- Stores update changes in the delta library as a new level.
- Continues based on the value in the GENERATE ELEMENT option. If this option is set to N, the product does not generate the element. If this option is set to Y, the product performs the following actions:
 - Reads the type definition for a Source Output Library specification, and then writes a copy of the current level of the element to that library. If the EXPAND INCLUDES option is set to Y, the product expands INCLUDE statements in the source.
 - Determines which processor group to use, and then executes the generate processor in that group (if one has been specified).
 - After the generate processor has been run for the element, updates the information in the Master Control File (MCF).
- Updates the Master Control File after the element is successfully updated. If the DELETE INPUT SOURCE option is set to Y, the product deletes the member indicated in the FROM library.

Updating a Member and Trailing Blanks in Input Files

When the product compares an input file against the source for an existing element that will be updated, *trailing blanks* (that is, blanks that occur in a data record after the last non-blank character) are ignored. This happens for comparison reasons when the base file is a fixed format, but the input file is variable. Variable includes not only MVS RECFM=V or VB files, but also HFS files and files sent from CA CM Enterprise Workbench.

Example: Use Trailing Blanks in Each Record

In this example, there is a small file that has trailing blanks in each record. In general, you will see trailing blanks in text records when the record size is fixed. Trailing blanks affect the compare logic performed by UPDATE ELEMENT logic in the product when it compares an input file to the existing element source.

```
'123  '  
'456  '  
'X    '
```

Example: Use One-record Element, Base File is Fixed

In this example, you have a one-record element in which the base file is a fixed record format with a record length of 80. The element data is a single record that contains "ABC" followed by 77 trailing blanks. You try to update this element with an input file with a record length of three with the content of "ABC." In this example, the product does not consider that the file has changed and does not create a new delta level.

Example: Use an Element on a Base File, Record Format is Variable

In this example, you have an element on a base file with the record format of variable. The element record is "ABC " and has one trailing blank. You try to update this element with an input file whose record is "ABC " (contains two trailing blanks). In this example, the product considers that the element has changed and creates a new delta level.

How Updating a Member Affects the Signout Status

Your site administrator can enable a signout capability on a system-by-system basis. When you allow the product to control a member when an element with the same name already exists in the target entry stage using the Update Elements action, the product performs the following actions and causes these effects:

- If the element exists up the map, a *fetch* is performed.
 - If your site administrator has enabled Signout Upon Fetch, there is no change to the signout status for the target element.
 - If your site administrator has not enabled Signout Upon Fetch, there is no change in the signout status for the target element.

Note: For more information about Signout Upon Fetch, see the *Administration Guide*.

How Updating a Member Affects CCIDs and Comments

When you allow the product to control a member when an element with the same name already exists in the target entry stage using the Update Elements action, you can enter a CCID and comment. When you enter a CCID and comment, the product sets the following options:

- Current Source CCID/Comment (if changed)
- Generate CCID/Comment (if generated)
- Last Action CCID/Comment
- Source Delta CCID/Comment (if changed)
- Component Delta CCID/Comment (if generate creates a delta)

Note: If you set the GENERATE ELEMENT option to N, the Update Elements action does not set the generate or component delta CCIDs and comments.

Update a Member

To control and manage a member when an element with the same name already exists in the target entry stage, you use the Update Elements action.

To update a member

1. Start the product using the instructions provided by your site administrator.
The Primary Options panel appears.
2. Enter **2** (Foreground) and press Enter.
The Foreground Options menu appears.
3. Enter **2** (Add/Update) and press Enter.
The Add/Update Elements panel appears.

Note: For more information about a field or option on the panel, press PF1.

Note: For more information about adding or updating an element with the Autogen option, see How Generating Elements with Autogen Works.

4. Choose *one* of the following steps.
 - a. To update a member, enter the information for the action, enter **U** in the Option field, and press Enter.
 - b. To update a member from the source library, enter the information for the action, leave the Option field blank and press Enter.

The Member Selection List appears. This list appears when the source data set is a library, you leave the Option field blank, the DISPLAY LIST option is set to **Y**, and when both the ELEMENT and MEMBER options are blank or have a name mask. The ISPF Locate command is supported on Member Selection lists.

5. (Optional) If the Member Selection List appears, enter **U** to the left of each member you want to update and press Enter.

The members are updated and you return to the Add/Update Elements panel. You can browse the listing data set for more information about each selected member.

Moving an Element Between Inventory Locations Using the Move Elements Action

To move an element from one inventory location (environment, stage) to the next location on a map route, you use the Move Elements action (available in foreground, batch, and in packages). Consider the following information when moving elements:

- You can move elements either with history or without history.
- You can only move elements from one environment to another if the elements start in Stage 2 of the source environment. For example, to move an element from Stage 1 of the Development environment into Stage 1 of the QA environment, you have to move the element to Stage 2 in Development, and then move it into Stage 1 in QA.

This section contains information for performing this development task in the *foreground* only.

Note: For more information about how to perform this element action in batch, see the *SCL Reference Guide*.

Another way to move an element is by using packages. Packages offer some advantages over the Move Elements action, such as being able to validate the package components and prevent elements from being moved until the elements have been assembled, compiled, and linked with current versions of all their dependencies. Promotion packages are a type of package that contain move actions only and the from environment and stage locations on all the move actions are the same.

Note: For more information about how to move elements using packages, see the *Packages Guide*.

Note: The Add, Move, Generate, and Signin actions can be performed from the Quick-Edit Option panel, if the CA Endeavor SCM Quick Edit option is enabled at your site. For more information about Quick-Edit, see the Quick Edit Option User Guide.

More information:

[How Moving Elements Between Inventory Locations Works](#) (see page 41)

[How Moving Elements with History Works](#) (see page 41)

[How Moving Elements Without History Works](#) (see page 43)

[How Moving Elements Affects the Signout Status](#) (see page 44)

[How Moving Elements Affects CCIDs and Comments](#) (see page 44)

[Move an Element](#) (see page 45)

Considerations When Moving Elements Using a Generate Processor

You can use move processors for the following reasons:

- To copy outputs (load modules and listings) from the source to the target stage.
- To recreate load modules at the target stage. This happens when a generate processor is used as a move processor. If you want to use a generate processor as a move processor, enter **G** in the PROCESSOR TO USE FOR THE MOVE ACTION option for the appropriate processor group.

Note: Do not recreate load modules at the target stage by coding a compile and link step in a move processor. If you do, the load module footprint does not sync with Master Control File (MCF) information for the element. Use the generate processor at the target stage, which will execute the compile and link and update the element's Master Control File.

How Moving Elements Between Inventory Locations Works

When you move an element from one inventory location (environment, stage) to the next location on a map route, the product performs as many of the following processes as necessary.

- *Source management*, which involves the actual move of the element and the Master Control File (MCF) updates.
- *Processor management*, which involves executing (at the target location) the Move processor of the source processor group or the Generate processor at the target processor group. This is determined by the processor group setting at the source location. Processing may include writing to a source output library and to the appropriate processor output libraries.
- *Delete processing*, which involves executing the Delete processor and deleting the element at the source location of the move.

Note: If a Move action fails, first respond to the error, and then resubmit your request. When you restart a Move action after a source management failure, the product performs source management, processor management, and delete processing. After a processor management failure, the product performs processor management and delete processing. After a delete failure, the product performs delete processing.

How Moving Elements with History Works

When you move an element with history using the Move Elements action, the product performs the following actions and causes these effects:

1. Determines if the element exists at the source and target location, if the ACKNOWLEDGE ELM JUMP option applies, if source management is necessary, and if processor management is necessary (meaning, a restart situation). If necessary, the product issues messages.

2. Performs *source management* by moving the element to the target stage. The processing that the product performs depends on whether the element exists at the TO location.
 - If the element does *not* exist at the target, the product performs the following steps:
 - Searches the map for subsequent occurrences of the element. If the element is found farther along the map, the base level of the element to be moved is compared with the current level of the subsequent element.
 - If the two levels are in sync, the product copies back (or *fetches*) the element up the map to the target location with all delta levels intact.
 - If the two levels are not in sync, the product issues a warning message and does not perform the copy (as long as the SYNC=N option was specified on the move action). If the SYNC=Y option was specified for the move action and the two are not in sync, the product issues a warning message and copies the element back to the target location and creates a sync level at the target. The new sync level reflects the differences between the base level of the source element and the current level of the fetched element.

Note: You can restrict the SYNC=Y option to prevent it from copying the element back to the target by activating the feature DO_NOT_SYNC_AT_TARGET=ON in the optional features table (ENCOPTBL).
 - Moves the element with the current level of the source element being moved to the target location, and the source element delta levels are appended to the target.
 - If the element is found at a stage that is not included in the map, a warning message is issued, and the search continues for the element in a stage that is included in the map.
 - If the element exists at the target location, the product performs level-matching. During level-matching, the product determines the sync point of the source and target elements by comparing the level timestamp of the base level of the source element with the current level of the target element. If the two timestamps do not match, the product checks the next oldest level of the source element, and so on.
 - If the sync point is found, the product moves the element from the FROM location to the TO location, appending the FROM location delta levels after the sync-point element.
 - If the two levels are different and SYNC=Y, the product first creates a sync level at the target reflecting the differences between the base level of the FROM element and the target, and then moves the element to the TO location and appends the FROM location delta levels to the target.
 - If the element does not exist at the target, the product moves the element from the source to the target location with all delta levels intact.

3. Completes source management after the element base and change levels have been moved by updating the Master Control File (MCF) to reflect the move.

Note: The value specified for Signout Upon Fetch (the SOFETCH parameter) in the Defaults table (C1DEFLT5) affects how the MCF for the element copied back will be updated. If Signout Upon Fetch is in effect, the element will be signed out to you unless it is already signed out to someone else. If Signout Upon Fetch is not in effect, the element will not be signed out to you.

4. Performs processor management by writing a copy of the element to a source output library, if one is defined for the associated element type at the target location. If EXPAND INCLUDES = Y (on the Type Definition panel), the product expands INCLUDE statements in the source.
5. Completes processor management by determining the processor group last associated with the element, and then executes the move or generate processor in that group if one has been specified. After the processor has been run for the element, the product updates the processor information in the Master Control file.
6. If the DELETE FROM ELEMENT option is set to Y, the product executes standard delete processing for the element.

How Moving Elements Without History Works

You can move an element without history using the Move Elements action by setting the WITH HISTORY option to N in your request. The product executes the request and attempts to find a sync level between the source and target elements beginning with the first level at the source and working forward through the deltas. When you move elements without history, the product performs the following actions:

- If a sync level is found, the elements are compared and a new level is created at the target that reflects the differences.
- If a sync level cannot be found and you set the SYNC option to N, a message is issued saying that the elements are "out of sync" and the MOVE action terminates.
- If a sync level cannot be found and you set the SYNC option to Y, an out-of-sync message is issued. The product then compares the last level of the source and last level of the target, and creates a new level at the target that reflects the differences.

How Moving Elements Affects the Signout Status

Your site administrator can enable a signout capability on a system-by-system basis. When you move an element from one inventory location (environment, stage) to the next location on a map route, the product performs the following actions and causes these effects:

- If the element exists up the map, a *fetch* is performed.
 - If your site administrator has enabled Signout Upon Fetch, the fetched (source) element (the element at the FROM location of the fetch) is signed out to you, if the element is not signed out to someone else. In addition, the target element is signed in.
 - If your site administrator has not enabled Signout Upon Fetch, the fetched (source) element is signed out to you, if the element is not signed out to someone else. In addition, the target element is signed in.
- If the element does *not* exist up the map, a fetch is not performed.
 - If your site administrator has enabled Signout Upon Fetch, the target element is signed in.
 - If your site administrator has not enabled Signout Upon Fetch, the target element is signed in.

Note: For more information about Signout Upon Fetch, see the *Administration Guide*.

How Moving Elements Affects CCIDs and Comments

When you move an element from one inventory location (environment, stage) to the next location on a map route using the Move Elements action (with history and without history), you can specify a CCID and comment. If you specify a CCID and comment, the product performs the following actions and causes these effects:

- If you use the Move Elements Action (with history):
 - The Source CCID/Comment is set from the start location value (Stage 1 value).
 - The Generate CCID/Comment is set from the start location value (Stage 1 value).
 - The Last Action CCID/Comment is set.
 - The Retrieve CCID/Comment is cleared.
 - The Source Delta CCID/Comment is carried with delta levels.
 - The Component Delta CCID/Comment is carried with delta levels.

- If you use the Move Elements Action (without history):
 - The Source CCID/Comment is set from the start location value (Stage 1).
 - The Generate CCID/Comment is set from the start location value (Stage 1).
 - The Last Action CCID/Comment is set.
 - The Retrieve CCID/Comment is cleared.
 - The Source Delta CCID/Comment is set from the last start location delta value.
 - The Component Delta CCID/Comment is set from the last start location delta value.

Move an Element

To move an element from one inventory location (environment, stage) to the next location on a map route, you use the Move Elements action.

Note: Another way to move an element is by using packages. Packages offer some advantages over the Move Elements action, such as being able to validate the package components and prevent elements from being moved until the elements have been assembled, compiled, and linked with current versions of all their dependencies. Promotion packages are a type of package that contain move actions only and the from environment and stage locations on all the move actions are the same. For more information about how to move elements using packages, see the *Packages Guide*.

To move an element

1. Start the product using the instructions provided by your site administrator.
The Primary Options panel appears.
2. Enter **2** (Foreground) and press Enter.
The Foreground Options menu appears.
3. Enter **5** (Move) and press Enter.
The Move Elements panel appears.
Note: For more information about a field or option on the panel, press PF1.
4. Specify the element you want to move, and enter a comment to explain the move.

5. Complete the following fields:

Sync

Enter **Y** (Yes) or **N** (No) to indicate whether you want the MOVE action performed when the base level of the element at the source location is different from the current level of the element at the target. When you enter **Y**, the product creates a "sync" level at the target that reflects the differences between the base level at the source location and the current level at the target. The move fails if these levels are different and you have specified SYNC = N.

Important! You must specify SYNC = Y when moving to a location where the out-of-sync condition exists. If the element does not exist at the target of the move, the SYNC=Y option searches up the map for the element and fetches (copies back) the element to the target and creates a sync level at the target. If you do not want this to happen, you must disable the DO_NOT_SYNC_AT_TARGET=ON option in the optional features table ENCOPTBL.

With History

Enter **Y** (Yes) or **N** (No) to indicate whether you want to move the element with history. When you move the element without history, the product searches through the element levels at the source location to find a matching level at the target location. The product then compares the two, and creates a new level at the target location that reflects the differences.

Signout To

Enter the TSO userid of the person at the target location for which you want to sign out the element. If the RETAIN SIGNOUT option is set to Y, you cannot use this option.

Delete from Element

Enter **Y** (Yes) or **N** (No) to indicate whether you want the product to delete the elements at the source location after moving them.

Important! If you enter N, you must specify SYNC = Y for any subsequent moves of this element.

6. Press Enter.

The element is moved to the next location on a map route.

Deleting Element Levels Using the Delete Elements Action

To delete all levels of an element from the product, any associated processors outputs from either stage, and if you use the CA Endeavor SCM Automated Configuration Manager (ACM), the component list for the element, use the Delete Elements action (available in foreground, batch, and in packages). If you use ACM, you can specifically delete only the component list for an element.

Note: This element action erases base and delta forms of an element and removes related information from the Master Control File (MCF) or component list.

This section contains information for performing this development task in the *foreground* only.

Note: For more information about how to perform this element action in batch, see the *SCL Reference Guide*. For more information about how to perform this element action using packages, see the *Packages Guide*. The Add, Move, Generate, and Signin actions can be performed from the Quick-Edit Option panel, if the CA Endeavor SCM Quick Edit option is enabled at your site. For more information about Quick-Edit, see the *Quick Edit Option User Guide*.

More information:

[Delete Element Levels](#) (see page 48)

How Deleting Element Levels Works

When you delete element levels using the Delete Elements action, the product performs the following actions and causes these effects:

- Verifies that the element currently exists in the stage you specify. If the element does not exist in the stage, you cannot delete the element.
- Determines whether the element is signed out to you. If the element is signed out to someone else, you must set the OVERRIDE SIGNOUT option to Y to delete the element.
- Deletes the element from its associated source output library, if one has been defined for the element type.
- Determines the processor group last associated with the element, and then executes the delete processor in that group if one has been specified.

- Deletes the element base level, and all change levels.
- Updates the Master Control File (MCF) to reflect this processing.

Note: When the Automated Configuration Manager (ACM) is installed, both the element and its component list are deleted.

Delete Element Levels

To delete all levels (base and delta members) of an element from the product, any associated processor outputs from either stage, and if you use the CA Endeavor SCM Automated Configuration Manager (ACM), the component list for the element, use the Delete Elements action.

To delete the levels for an element

1. Start the product using the instructions provided by your site administrator.

The Primary Options panel appears.

2. Enter **2** (Foreground) and press Enter.

The Foreground Options menu appears.

3. Enter **6** (Delete) and press Enter.

The Delete Elements panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. Specify the element you want to delete, enter the additional information for the action, enter **#** (pound sign) in the Option field, and press Enter.

Note: If you enter a CCID, it is logged to IBM's System Management Facilities (SMF), if enabled, and is available to exits in the product.

5. (Optional) If the System Selection List, Subsystem Selection List, and Element Selection List appears, enter **S** to the left of the system, subsystem, and element you want to delete.

The element levels are deleted, and the Master Control File (MCF) is updated.

Creating Outputs Using the Generate Elements Action

To create an executable form of an element (that is, outputs: object modules, load modules, listing, and so forth), use the Generate Elements action (available in foreground, batch, and in packages). You determine the kinds of outputs this element action produces in the generate processors that you write for each type in your software inventory.

The product can generate elements in two ways:

- Automatically. This happens when you add, update, transfer, or restore elements (unless you have set the GENERATE ELEMENTS option to N).
- Explicitly, when you execute an explicit GENERATE element action.

The Copyback and NoSource action options determine what element source is used to generate elements. They are mutually exclusive and both require the option Build Using Map. The Autogen option improves processing by eliminating duplicate processing of components and reduces the work required of users, who no longer need to use the Automated Configuration Manager Query facility (ACMQ) to create additional Generate actions for element components and then run another batch job to process them.

Brief descriptions of how each of these options affect processing are provided next.

Copyback

If you select this option, CA Endeavor SCM first copies the current level of the element back to the FROM stage, then generates the element. CA Endeavor SCM searches for the element first in the current environment, then in other stages along the map. If the element currently exists in the FROM stage, CA Endeavor SCM ignores the COPYBACK option and simply generates the element.

NoSource

If you select this option, when the target location has a sourced element, the element is generated in place. When the target location has a sourceless element, the element is generated at the target location using the source of the first occurrence of the element found up the map. When the element does not exist at the target location, the element is generated at the target location using the source of the first occurrence of the element found up the map. The source is not fetched to the target.

Autogen

If you select this option for an element, the elements that use this component element are automatically generated. For example, specifying Autogen for an element of Type Macro automatically generates the source elements that use the macro, which then generates the appropriate LNK elements. Autogen is available in batch only for the Add, Update, and Generate actions and cannot be used in packages.

An element that uses a component element is known as a *using element*. For example, if Autogen is specified for copybook COPYA, then the programs that use that copybook are using elements. Autogen generates *only* those using elements that are in the same logical map as the target component. Autogen Span options *also* generates using elements that are found in different Systems and Subsystems within the Environment and Stage of the logical map.

Note: For more information about Autogen and the Span options, see the scenario "How to Automatically Generate *Using Elements*" in the *Scenario Guide*.

This section contains information for performing this development task in the *foreground only*.

Note: For more information about how to perform this element action in batch, see the *SCL Reference Guide*. For more information about how to perform this element action using packages, see the *Packages Guide*. The Add, Move, Generate, and Signin actions can be performed from the Quick-Edit Option panel, if the CA Endeavor SCM Quick Edit option is enabled at your site. For more information about Quick-Edit, see the *Quick Edit Option User Guide*.

More information:

[How Generating Elements Without Copyback Works](#) (see page 52)

[How Generating Elements Affects the Signout Status](#) (see page 53)

[How Generating Elements Affects CCIDs and Comments](#) (see page 54)

[How Generating Elements with Copyback Works](#) (see page 51)

[How Generating Elements with NoSource Works](#) (see page 52)

Generate an Element Output

To create an executable form of an element (that is, outputs: object modules, load modules, listing, and so forth), use the Generate Elements action.

Important! Do not execute a stand-alone GENERATE action against a load module. If you do, change regression may happen because generate processors update the output library, not the actual source. If you want to update a load module, perform an ADD or UPDATE action with the GENERATE ELEMENT option set to Y.

To generate an element output

1. Start the product using the instructions provided by your site administrator.

The Primary Options panel appears.

2. Enter **2** (Foreground) and press Enter.

The Foreground Options menu appears.

3. Enter **4** (Generate) and press Enter.

The Generate Elements panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. Specify the element for which you want to generate an output, enter the additional information for the action, enter **S** in the Option field, and press Enter.

5. (Optional) If the System Selection List, Subsystem Selection List, and Element Selection List appears, enter **S** to the left of the system, subsystem, and element for which you want to generate an output.

The output is generated for the element.

How Generating Elements with Copyback Works

Generate an element with copyback when you change an element and you want to recompile the affected programs. For example, you change a copybook named Copy that is shared by programs PROGX, PROGY, and PROGZ. When you generate an element with copyback using the Generate Elements action, the product performs the following actions and causes these effects:

1. Searches beyond the current stage, in every stage in the current and subsequent environments in the map.
2. If the product finds the element, copies the current version of the element to the target stage, along with the processor group name last used for the element. Consider the following options when generating elements:
 - To restrict the search to the current environment, generate the element by setting the BUILD USING MAP option to N.
 - If the product finds the element in a stage that is not included in the map, a warning message is issued, and the search continues for the elements in a stage that is included in the map.

3. Updates the Master Control File (MCF).

Note: The Signout Fetch (SOFETCH) value in the Defaults Table (C1DEFULTS) determines how the MCF is updated for the element copied back (fetched). If Signout Upon Fetch is in effect, the element is signed out to you unless it is already signed out to someone else. If Signout Upon Fetch is not in effect, the element is not signed out to you.

4. Checks the element type definition for a source output library specification, and then writes a copy of the current level of the element to that library. If the EXPAND INCLUDES option is set to Y on the Type Definition panel, the product expands INCLUDE statements in the source.
5. Determines the processor group, and then executes the processor in that group, if one has been specified.
6. After the generate processor is run for the element, updates the information in the Master Control File (MCF).

How Generating Elements Without Copyback Works

When you generate an element without copyback using the Generate Elements action, the product performs the following actions and causes these effects:

1. Determines whether the element is signed out to you.
2. Checks the element type definition for a source output library specification, and then writes a copy of the current level of the element to that library. If the EXPAND INCLUDES option is set to Y on the Type Definition panel, the product expands INCLUDE statements in the source.
3. Determines the processor group, and then executes the processor in that group, if one has been specified.
4. After the generate processor is run for the element, updates the information in the Master Control File (MCF).

How Generating Elements with NoSource Works

NoSource is an option for the Generate action that eliminates the need to fetch an element's source back to the target location when the element does not already exist at the target. The action options Copyback and NoSource are mutually exclusive; if one is set to Y, the other must be set to N. The list option Build Using Map must be set to Y.

Assuming that the element source is not at the target location and the element source is located up the map from the target, the effect of the Generate NoSource action is as follows:

1. The element's source is not fetched back to the target location from up the map.
2. The first occurrence of the element up the map from the target location is used as input to the generate processor. This element source is used as follows:
 - a. All target inventory C1 symbols are set with the element's source inventory fields before doing the C1BASELIB substitution.
 - b. The C1BASELIB symbol is set with the library base data set defined at the source location (up the map).
 - c. After the C1BASELIB symbol is resolved, the target inventory C1 symbol is reset to the action's target inventory.
3. After the Generate action completes, the targeted location coded in a Generate action contains the outputs created by the generate processor. The MCF element created at the target location is identified as a sourceless element and contains data similar to a fetched back element except that the element base and delta name fields are blank and the record is marked as a sourceless element.

Note: Elements with forward deltas must use the CONWRITE utility in their processors to rebuild the element source. This utility supports sourceless elements by using the actual source location from up the map to rebuild the element. CONWRITE also supports forward and reverse deltas.

How Generating Elements Affects the Signout Status

Your site administrator can enable a signout capability on a system-by-system basis. When you create an executable form of an element using the Generate Elements action, the product performs the following actions and causes these effects:

- If the element exists up the map (with copyback), a *fetch* is performed.
 - If your site administrator has enabled Signout Upon Fetch, the fetched (source) element is signed out to you, if the element is not signed out to someone else. In addition, the target element is signed out to you.
 - If your site administrator has not enabled Signout Upon Fetch, there is no change in the signout status, and the target element is signed out to you.

- If the element does *not* exist up the map (without copyback), a fetch is not performed.
 - If your site administrator has enabled Signout Upon Fetch, there is no change in the signout status of the target element.
 - If your site administrator has not enabled Signout Upon Fetch, there is no change in the signout status of the target element.

Note: For more information about Signout Upon Fetch, see the *Administration Guide*.

How Generating Elements Affects CCIDs and Comments

When you create an executable form of an element using the Generate Elements action (with copyback and without copyback), you can enter a CCID and comment. When you enter a CCID and comment, the product sets the following options:

- When you generate an element *with* copyback, the product sets the following options:
 - Current Source CCID/Comment (to the copied back value, or the Stage 2 value)
 - Generate CCID/Comment
 - Last Action CCID/Comment
 - Source Delta CCID/Comment (to the copied back value, or the Stage 2 value)
 - Component Delta CCID/Comment
- When you generate an element *without* copyback, the product sets the following options:
 - Generate CCID/Comment
 - Last Action CCID/Comment
 - Component Delta CCID/Comment (if generating creates a delta)

Printing Element Information Using the Print Elements Action

When performing your development tasks, you can print element information that the product has collected using the Print Elements action (available in foreground and batch). You can print the following element information:

- All statements at a specific element level, as well as the level at which each statement was inserted.
- All changes (insertions and deletions) made to the element at a specific level.

- A history of all statements in all levels of the element.
- A one-line summary of information for each level.
- Master Control File (MCF) information.
- Input component information extracted from the ACMQ files (ACMROOT and ACMXREF) for the specified element. The Explode option prints this information. When Explode is specified, the Print action for the element jumps to the top of the list to be executed. Therefore, any Print actions that use the Explode option are processed first before other actions are processed in Type sequence order.

Note: If you want to send your printed element information to a data set, you must follow the data set requirements in the *SCL Reference Guide*.

This section contains information for performing this development task in the *foreground* only.

Note: For more information about how to perform this element action in batch, see the *SCL Reference Guide*.

Print Element Information

When performing your development tasks, you can print element information that the product has collected.

To print element information

1. Start the product using the instructions provided by your site administrator.
The Primary Options panel appears.
2. Enter **2** (Foreground) and press Enter.
The Foreground Options menu appears.
3. Enter **7** (Print) and press Enter.
The Print Elements panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. Choose *one* of the following steps.

Note: By default, the product prints the current level of the element. If you want to print a level that precedes the current level, first use option **S** to display a Summary of Levels for the element, and then select the level you want to print from that display.

- a. To display an Element Selection List and refine your print request, leave the Option field blank and press Enter.
- b. To print all statements at a specific element level, as well as the level at which each statement was inserted, enter the information for the element, enter **P** in the Option field, and press Enter.
- c. To print all changes (insertions and deletions) made to the element at a specific level, enter the information for the element, enter **PC** in the Option field, and press Enter.
- d. To print a history of all statements in all levels of the element, enter the information for the element, enter **PH** in the Option field, and press Enter.
- e. To print a one-line summary of information for each level, enter the information for the element, enter **PS** in the Option field, and press Enter.
- f. To print Master Control File (MCF) information, enter the information for the element, enter **PM** in the Option field, and press Enter.
- g. To print a component explode report showing all the components of the element, enter the information for the element, enter **PX** in the Option field, and press Enter.
- h. To print an element's current output listing, enter the information for the element, enter **PL** in the Option field, and press Enter.

The appropriate SCL for your print request is saved in a temporary data set. When you exit the product, you can print the information.

Removing the User Signout Using the Signin Elements Action

To remove the signout user ID associated with an element, you use the Signin Elements action (available in foreground, batch, and in packages). SIGNIN is only available for systems where signin and signout is in effect.

This section contains information for performing this development task in the *foreground* only.

Note: For more information about how to perform this element action in batch, see the *SCL Reference Guide*. For more information about how to perform this element action using packages, see the *Packages Guide*. The Add, Move, Generate, and Signin actions can be performed from the Quick-Edit Option panel, if the CA Endeavor SCM Quick Edit option is enabled at your site. For more information about Quick-Edit, see the *Quick Edit Option User Guide*.

More information:

[How Removing the User Signout Affects CCIDs and Comments](#) (see page 57)

[How Removing the User Signout Affects the Signout ID](#) (see page 58)

[Remove the User Signout](#) (see page 58)

How Removing the User Signout Works

When you remove the signout user ID associated with an element using the Signin Elements action, the product performs the following actions and causes these effects:

- Determines whether the element is signed out to you. If the element is signed out to someone else, you must set the OVERRIDE SIGNOUT option to Y to sign in the element.

Note: Your site administrator must set up permissions for you to use the OVERRIDE SIGNOUT option.

- Updates the Master Control File (MCF) for the element by removing the current signout (that is, the user ID associated with the last RETRIEVE action).

Note: When you sign in an element, you can assign the element to another person using the SIGNOUT TO option.

How Removing the User Signout Affects CCIDs and Comments

When you remove the signout user ID associated with an element, the product clears the Retrieve CCID/Comment for the element.

How Removing the User Signout Affects the Signout ID

Note: You cannot sign in an element unless it is already signed out to you, unless you set the OVERRIDE SIGNOUT option to Y.

Your site administrator can enable a signout capability on a system-by-system basis. When you remove the signout user ID associated with an element using the Signin Elements action, the product performs the following actions and causes these effects:

- Clears the SIGNOUT ID field at the stage at which the signin is being performed, unless you also specify a user ID in the SIGNOUT TO field.
- If you specify a user ID in the SIGNOUT TO field, the product sets the SIGNOUT ID at the stage at which the action is being performed to the user ID specified in the action.

Remove the User Signout

If an element is currently signed out to your user ID, you can sign in the element, which removes the signout ID associated with the element. In addition, after the element is signed in, you can sign out the element to a different user ID. If your site administrator has set up authorization for you, you can sign in an element that is not signed out to you.

To remove the user signout

1. Start the product using the instructions provided by your site administrator.
The Primary Options panel appears.
2. Enter **2** (Foreground) and press Enter.
The Foreground Options menu appears.
3. Enter **8** (Signin) and press Enter.
The Signin Elements panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. Specify the element you want to sign in and enter the additional information for the action. Enter **SI** in the Option field, and press Enter. The following actions are optional:

Override Signout

Specifies whether element signin is allowed when the element is not currently signed out to you. Acceptable values are:

Y – Sign in the element even if it is not currently signed out to you. Your site administrator must set up permission for you to use to specify Y.

N – Default. Do not allow the SIGNIN action unless the element is already signed out to

you.

Signout To

Specifies the user ID that the element will be signed out to after it is signed in.

5. (Optional) If the System Selection List, Subsystem Selection List, and Element Selection List appears, enter **S** to the left of the system, subsystem, and element you want to signin.

The element is signed in. If you entered a user ID in the Signout To field, the element is signed out to that user ID.

Chapter 4: Packaging Element Actions

This section contains the following topics:

[Packages](#) (see page 61)

[Create a Package](#) (see page 61)

[Cast a Package](#) (see page 62)

[Review and Approve a Package](#) (see page 63)

[Execute a Package](#) (see page 64)

[Submit a Package for Batch Processing](#) (see page 66)

Packages

When you need to group a set of actions that require approval before execution, you *package* the elements together. You use the following workflow when working with packages.

1. [Create a Package](#) (see page 61)
2. [Cast a Package](#) (see page 62)
3. [Review and Approve a Package](#) (see page 63)
4. [Execute a Package](#) (see page 64)
5. [Submit a Package](#) (see page 66)

Important! Before you work with packages, it is recommended that you read the *Packages Guide* to ensure that you are familiar with the package life cycle and how to correctly process packages.

Create a Package

You create a package by first defining the package, and then including SCL in the package to specify element actions to be performed. Packages can be executed online or submitted in batch.

To create a package

1. Start the product using the instructions provided by your site administrator.
The Primary Options panel appears.
2. Enter **5** (Package) and press Enter.
The Package Options menu appears.

3. Enter a name for the package, type **2** (Create/Modify) in the Option field and press Enter.

The Create/Modify Package panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. Select the package type in the Promotion Package field. Enter Y, if this is a promotion package, or N if this is not a promotion package.

This option determines whether the package will be processed as a promotion package.

5. Enter **B** in the Option field to open the online SCL generator.

6. Enter the package information and press Enter to submit the build request.

The SCL Generation panel appears.

7. Specify the actions in the Option field to build the package and press Enter.

Note: These actions are placed in the request data set displayed at the bottom of the page.

The appropriate action panel appears.

8. Press END to return to the Create/Modify Package panel.

The package is created, and can now be cast.

Cast a Package

Some packages must be reviewed and approved before being executed. Casting a package is the first step in the package review process.

To cast a package

1. Start the product using the instructions provided by your site administrator.

The Primary Options panel appears.

2. Enter **5** (Package) and press Enter.

The Package Options menu appears.

3. Enter the package name, leave the PACKAGE ID field blank to display a Package Selection List, or enter a *name mask*. Enter **3** (Cast) in the OPTION field and press Enter.

The Cast Package panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. If the Package Selection List appears, select the package you want to cast and press Enter.

The Cast Package panel appears.

5. (Optional) Enter **S** in the Option field and press Enter to view the package SCL before you cast the package.

The Display SCL panel appears.

6. (Optional) Review the SCL and press END.

The Cast Package panel appears.

7. If the Cast Package panel appears, enter **C** in the Option field to cast the package.

Note: After you cast a package, you cannot edit it.

Specify the information in the VALIDATE COMPONENTS, ENABLE BACKOUT, and EXECUTION WINDOW fields.

8. Press Enter to submit the cast request.

9. Press END to return to the Package Options menu.

When you cast a package that requires approval, the package status is changed to In-approval. When you cast a package that does not require approval, the package status is changed to Approved. After a package has been cast, it can be approved and then executed.

Review and Approve a Package

If one or more *approver groups* are associated with the inventory areas included in a package, then the package must be reviewed and approved before it can be executed. To be approved, the package must be accepted by all of the required approvers, be accepted by a quorum of approvers, and not be denied by any approver.

To review a package

1. Start the product using the instructions provided by your site administrator.

The Primary Options panel appears.

2. Enter **5** (Package) and press Enter.

The Package Options menu appears.

3. Enter the package name, leave the PACKAGE ID field blank to display a Package Selection List, or enter a *name mask*. Enter **4** (Review) and press Enter.

The Review Package panel appears or the Package Selection List appears.

Note: For more information about a field or option on the panel, press PF1.

4. If the Package Selection List appears, select the package you want to review and press Enter.
The Review Package panel appears.
 5. On the Review Package panel, choose *one* of the following steps.
 - a. To review the SCL for the package, leave the Option field blank and press Enter.
The Summary SCL panel appears.
 - b. To approve the package, enter **A** in the Option field and press Enter.
The Package Options menu appears with a message saying that the package has been approved.
 - c. To deny the package, enter **D** in the Option field and press Enter.
The Package Options menu appears with a message saying that the package has been denied.
 - d. To view the approver groups for the package, enter **L** in the Option field and press Enter.
The Approver Groups panel appears. Press END to return to the Package Review panel.
 - e. To add or change notes associated with the package, enter **N** in the Option field and press Enter.
The Package Note Text panel appears. Press END to return to the Package Review panel.
- The package is reviewed and approved, and can be executed.

Execute a Package

After a package has been approved, you can submit it for execution now or for batch processing. Before executing the package, the product validates the package to make sure you have the authority to perform the package's actions, the execution request is within the execution window, and that no elements have changed since the package was cast.

To execute a package

1. Start the product using the instructions provided by your site administrator.
The Primary Options panel appears.
2. Enter **5** (Package) and press Enter.
The Package Options menu appears.

3. Enter **5** (Execute) and press Enter.

The Execute Package panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. Enter the package name, leave the PACKAGE ID field blank to display a Package Selection List, or enter a *name mask*, and press Enter.

The Execute/Submit Package panel or the Package Selection List appears.

5. If the Package Selection List appears, select the package you want to execute and press Enter.

The Execute/Submit Package panel appears.

6. On the Execute/Submit Package panel, enter **E** in the OPTION field.

The package is executed.

Note: You can submit the job for batch processing from the Execute/Submit panel, by entering **S** in the OPTION field.

Note: For more information about batch submission, see Submit a Package for Batch Processing.

Submit a Package for Batch Processing

You can submit a package to be executed in batch. Batch processing lets you request concurrent action processing.

To submit a package for batch processing

1. On the Execute/Submit Package panel, type **S** in the OPTION field and press Enter.
The Submit Package panel.

2. Edit the options.

Include JCL

Y--Additional JCL is included when the job is submitted.

N--No additional JCL is included.

Concurrent Action Processing

Use this field to indicate whether or not you want to use concurrent action processing. Valid values are Y and N. The default value when you enter the panel is N. If this feature is not enabled for your site, this option is read-only.

Note: The *Concurrent Action Processing* facility causes certain element action requests to be executed concurrently, which reduces the elapsed time it takes to process multiple actions. Before you can use this facility, it must be enabled by your CA Endeavor SCM administrator.

Concurrent Number

Specify the number of concurrent actions to be processed, if you are using concurrent action processing. The default is the SPAWNCNT value set in C1DEFLT5. If you overwrite the default and then decide you want to use the default, either type in the default value or blank out this field. Valid values are 02 through the Max number shown on the panel. The Max number is the value of SPAWNMAX specified in the C1DEFLT5.

3. Optional. Type **E** in the OPTION field and press Enter.
The JCL to Be Included with Batch Job panel opens.
4. Optional. On the JCL to Be Included with Batch Job panel, enter additional JCL to be included with the job.
This JCL is included when the job is submitted, if the Include JCL option on this Submit panel is set to yes.
5. Press the End key.
The Submit Package panel reopens.
6. Type **S** in the OPTION field and then press Enter.
The package is submitted for batch execution.

Chapter 5: Performing Batch Tasks in Foreground

This section contains the following topics:

[Perform Batch Tasks in Foreground](#) (see page 67)

Perform Batch Tasks in Foreground

CA Endeavor SCM simplifies batch tasks by helping you generate SCL and then wrapping the SCL for you with the appropriate JCL. The SCL is a member in a request data set. When you submit the job, the JCL skeleton is added and the job is submitted for processing. This method is available when you select option 3, Batch, from the Primary Options Menu.

As an alternative to performing batch tasks in foreground, you can perform batch tasks from the ISPF Menu. You can write your own JCL, with embedded SCL, and the job stream is a member in a data set. To run the job, you access the member from the ISPF Menu using the options 1 (View) or 2 (Edit) and then type SUB on the command line and press Enter to submit the job.

Note: For more information on writing your own SCL commands, see the SCL Reference Guide.

Note: For more information on performing package processing in batch, see the Packages Guide.

Build Element Action SCL in Foreground for Execution in Batch

You can build SCL actions in foreground and either submit them directly from foreground or schedule them for later execution.

The actions involved are the same as foreground, plus Archive, List Element, List Member, Print Member, Transfer, and Validate. However, the Restore action can only be performed in batch from the ISPF Menu. Restore is not available on the SCL Generate foreground panel.

To build element action SCL in foreground for execution in batch

1. Allocated the partitioned or sequential data set you plan to use.
2. On the Primary Options Menu, select option **3**, Batch.

The Batch Options Menu opens.

3. Define the data set name in the Request Data Set fields. Specify the data set to which you want to write the action requests. This data set must be a partitioned data set or a sequential file, and must be allocated prior to referencing it on this panel. As an alternative, you can use the OTHER PARTITIONED OR SEQUENTIAL DATA SET field.
4. Use the following fields to enter additional information:

Append

Use this field to indicate whether you want to add new requests to the end of an existing data set or library member.

Y-- Add new requests to the end of an existing data set or library member.

N-- Default. Overwrite any data currently in the data set or library member.

Include JCL

Use this field to indicate whether you want to include JCL in addition to the standard execution JCL. Acceptable values are:

N-- Default. Do not include additional JCL.

Y-- Include additional JCL. This JCL must already have been defined. This is used in conjunction with Option 5 - Build JCL.

STOPRC

Use this field to insert a SET STOPRC value. Valid values range from 4 to 16. Changing the STOPRC value on the panel will override the system default value of 16. CA Endeavor SCM will stop processing actions when a previous action has a return code equal to or higher than the defined STOPRC.

Note: If you append SCL to existing SCL and the existing SCL already contains a SET STOPRC card, then this option will not modify the existing SET STOPRC card.

Other Partitioned or Sequential Data Set

Use this field to define the data set by entering the appropriate data set name (and member, if a library) in the following format: 'dataset(member)'.

Job Statement Information

These fields allow you to provide jobcard settings for submitting batch jobs. Provide all jobcard parameters, including job name, time, message parameters, remote print information, and any other standards in use at your site.

5. Select option **1**, BUILD SCL and press Enter.

The SCL Generate panel opens. This panel allows you to select the type of action request you want to generate, or to request an element display. The request data set and append information defined on the Batch Options Menu appear at the bottom of the screen.

6. Specify the option number that corresponds to the action you wish to perform and then press Enter. Choose from the following options:

1 Display

Display element information before writing action requests for the element.

2 Add/Update

Add or update elements.

3 Retrieve

Copy elements to a user data set.

4 Generate

Generate elements.

5 Move

Move elements from one map location to another.

6 Delete

Remove elements and/or element component lists from either stage.

7 Print Element

Print any of several detailed element reports showing element source, the history of the element, changes made to the element, summary of levels for the element, or Master Control File information about the element.

8 Signin

Remove signout IDs from elements.

9 Transfer

Move elements from a map location to a location not on the map.

10 Print Member

Print or browse footprinted members from a library.

11 List Element

List elements from the Master Control File, where the list takes the form of action requests.

12 List Member

List or browse footprinted members from a library. Again, the list takes the form of action requests.

13 Archive

Write the current version of elements to a sequential file (known as an archive data set), generally deleting it following the archive.

14 Validate

Perform element master, synchronization, and component validation against the element you specify.

After you press Enter, the action panel opens that corresponds to the action option you selected.

Edit Batch Requests

You can edit existing request data sets from foreground.

To edit an existing request data set

1. From the Primary Options Menu, select option **3**, BATCH.

The Batch Options Menu opens.

2. On the Batch Options Menu, specify the data set you want to edit using the REQUEST DATA SET or OTHER PARTITIONED OR SEQUENTIAL DATA SET fields.
3. On the Batch Options Menu, select option **2**, EDIT and press Enter.

An edit panel showing the contents of the specified data set opens. The edit panel uses the standard ISPF Edit facility.

4. Use the edit panel to review the action requests in the data set, or to modify the data set.

Note: For more information about coding the batch requests, see the SCL Reference Guide.

Note: The ISPF Edit Recovery facility is not available for this function. If your TSO session crashes, then any changes made, but not saved, will be lost. You can, however use the ISPF/PDF Edit RECOVERY ON/OFF command to control whether a recovery data set is associated with the element. The RECOVERY ON command is used in conjunction with the UNDO Edit command.

Validate Batch Jobs

You can validate the SCL in an action request data set before submitting the job. If a job statement is specified in the JOB STATEMENT INFORMATION fields on the Batch Options Menu, it is also validated.

To validate a batch job

1. On the Batch Options Menu, specify the data set you want to validate using the REQUEST DATA SET or OTHER PARTITIONED OR SEQUENTIAL DATA SET fields. If you want to specify a job statement, enter this information in the JOB STATEMENT INFORMATION fields on the Batch Options Menu.
2. Select option 4, VALIDATE and press Enter.

The action request data set and jobcard are validated. The syntax of the action requests is checked as if the job was going to run, and any appropriate error messages are returned. The ISPF/PDF Browse facility returns a panel that shows any syntax errors within SCL statements defined in the REQUEST DATA SET.

Build Additional JCL

You can define JCL to be included with other batch requests, for example, if an action request references a source or target file by DDname, or identifies an archive data set.

To build additional JCL

1. From the Batch Options Menu, select option 5, BUILD JCL, and press ENTER.
The JCL To Be Included In Batch Job panel opens.
2. Fill in the JCL (using complete statements) that you want to include with the batch job.
3. Press PF3 to return to the Batch Options Menu.

Submit Batch Job for Processing

You can submit batch jobs for processing.

Note: Before submitting a job, you can validate the request data set and specify DD statements to be included with the job using the VALIDATE and BUILD JCL options on the Batch Options Menu.

To submit a job that executes action requests in batch

1. On the Batch Options Menu, specify the data set containing the actions that you want to process in either the REQUEST DATA SET or OTHER PARTITIONED OR SEQUENTIAL DATA SET fields, as described next.

Request Data Set

Use these fields to identify the request data set you want to submit for execution. Alternatively, use the OTHER PARTITIONED OR SEQUENTIAL DATA SET field.

Other Partitioned or Sequential Data Set

Data set name (and member name if the data set is a library) of the request data set you want to submit for execution: 'data set (member name)' format.

Include JCL

Indicate whether you want to include JCL in addition to the standard execution JCL: Y (yes) or N (no). If you specify Y, you must have defined the additional JCL already, using option 5. The default is N.

Job Statement Information

Jobcard to be submitted with the execution.

Concurrent Action Processing

Use this field to indicate whether or not you want to use concurrent action processing. Valid values are Y and N. The default value when you enter the panel is N. If this feature is not enabled for your site, this option is read-only.

Concurrent Number

Specify the number of concurrent actions to be processed. The default is the SPAWNCNT value set in C1DEFLT5. If you overwrite the default and then decide you want to use the default, either type in the default value or blank out this field. Valid values are 02 through the Max number shown on the panel. The Max number is the value of SPAWNMAX specified in the C1DEFLT5.

2. Select option **3**, SUBMIT and press Enter.

The JCL to submit the job is dynamically built, including the jobcard from the Batch Options Menu, a standard job stream provided during installation, and any additional JCL. TSO displays a message similar to the following example to let you know the job has been submitted:

```
IKJ56250I JOB ZSXJMA1A(JOB02433) SUBMITTED
```

```
***
```


Chapter 6: Viewing Inventory Information

This section contains the following topics:

- [Inventory Components](#) (see page 75)
- [View Element Information](#) (see page 76)
- [View Footprint Information](#) (see page 81)
- [View Site Definitions](#) (see page 82)
- [View Stage Information](#) (see page 82)
- [View System Definitions](#) (see page 83)
- [View Subsystem Definitions](#) (see page 83)
- [View Type Definitions](#) (see page 84)
- [View Processor Group Definitions](#) (see page 85)
- [View Approver Group Definitions](#) (see page 86)
- [View Approver Group Relationship Definitions](#) (see page 87)
- [View Environment Information](#) (see page 88)
- [View Site Symbolics](#) (see page 88)
- [View Global Type Processing Sequence](#) (see page 89)

Inventory Components

When your *inventory components* are set up, information about each component is collected and saved. The process that you use to view inventory component information is similar for each component. You enter the information associated with the component you want to view (for example, environment, system, subsystem, element, type, and stage) and then view the information that you need.

View Element Information

You can view detailed information about each element in your software inventory. You can view a summary of changes made to an element, element information from the Master Control File (MCF), all statements in the current level of an element (and the level at which each statement was inserted), all changes (insertions and deletions) made to an element up to the current level, and all statements in all levels of an element, from the base level to the current level (including the level at which each insertion and deletion occurred). If you want to view information about more than one element, or you cannot fully qualify the element name, see the procedure View Element Information from a Selection List.

The options to view information about sourceless elements and elements of log delta format are limited.

To view element information

1. Start the product using the instructions provided by your site administrator.

The Primary Options panel appears.

2. Enter 1 (Display) and press Enter.

The Display Options menu appears.

Note: For more information about a field or option on the panel, press PF1.

3. Enter 1 (Element) and press Enter.

The Display Elements/Component List panel appears.

Note: The Display Elements/Component Lists panel appears for CA Endeavor SCM ACM users. If you do not have CA Endeavor SCM ACM at your site, the Display Elements panel appears and you cannot view component list information.

4. Enter the information associated with the element you want to view (environment, system, subsystem, element, type, and stage).

5. Make sure all the List Options are set to N.

6. Enter your selection in the OPTION field and press Enter.

Depending on the option you selected, you get the following results.

S--To view a summary of changes. This option is not valid for sourceless elements.

The Summary of Levels panel opens. You can then view additional information for one or more of the levels by entering a value to the left of the level and pressing Enter. (Enter B for the Element Browse panel, enter C for the Element Changes panel, or enter H for the Element History panel).

M--To view element information from the Master Control File (MCF).

The Element Master panel opens showing MCF information for the element.

B--To view all statements in the current level. This option is not valid for sourceless elements.

The Element Browse panel opens showing all of the statements in the element level and the level at which each statement was inserted. By default, the current level is displayed. To view previous levels, navigate to the Summary of Levels panel, enter B to the left of each level you want to view, and press Enter. For elements of log delta format, only the current (last) level can be browsed, and the current level does not show the source code changes, because source code changes are not kept for elements of log delta format.

C--To view all changes (insertions and deletions). This option is not valid for sourceless elements or elements of log or image delta format.

The Element Change panel opens showing all inserts and deletions made to the element between the specified level and its immediate predecessor. By default, the current level is displayed. To view the changes from previous levels, navigate to the Summary of Levels panel, enter C to the left of each level you want to view, and press Enter.

H--To view all statements in all levels. This option is not valid for sourceless elements or elements of log or image delta format.

The Element History panel opens showing all statements in all levels of the element, from the base level through the current level. You can view the level at which each insertion and deletion occurred. By default, the current level is displayed. To view the history of previous levels, navigate to the Summary of Levels panel, enter H to the left of each level you want to view, and press Enter.

View Element Information from a Selection List

If you want to view information about more than one element, or if you cannot fully qualify the element name, you can view element information from a selection list.

The options to view information about sourceless elements and elements of log delta format are limited.

To view element information from a selection list

1. Start the product using the instructions provided by your site administrator.

The Primary Options panel appears.

2. Enter **1 (Display)** and press Enter.

The Display Options menu appears.

Note: For more information about a field or option on the panel, press PF1.

3. Enter **1 (Element)** and press Enter.

The Display Elements/Component List panel appears.

Note: The Display Elements/Component Lists panel appears for CA Endeavor SCM ACM users. If you do not have CA Endeavor SCM ACM at your site, the Display Elements panel appears and you cannot view component list information.

4. Enter the information associated with the element you want to view (environment, system, subsystem, element, type, and stage) and set the following list options.

Use the first three list options to limit the elements shown in the Element Selection List. If you select Apply List Filters, the Apply List Filters panel opens, which lets you further filter the list. If you do not know the system or subsystem name, you can wildcard these fields and set the Display System/Subsys List to Y to display the System Selection List or Subsystem Selection Lists panels.

Display List

Indicates whether you want to use list panels. Select **Y** for yes, or **N** for no. The default is N.

Apply List Filters

If you set the list option Apply List Filters to Y, then the Apply List Filters panel opens and shows the following filter options:

Where CCID EQ

Specify a CCID to limit the list to those elements whose current, last action, generate, or retrieve CCID match the CCID specified here. You can use a wildcard in this field. You can further limit the list by eliminating matches for any of the following categories: current, last action, generate, or retrieve.

Where User ID

Specify a user ID to limit the list to those elements whose current, last action, generate, or retrieve user ID match the user ID specified here. You can use a wildcard in this field. You can further limit the list by eliminating matches for any of the following categories: current, last action, generate, or retrieve.

Where Processor Group EQ

Specify a processor group name to limit the list to those elements to which the specified processor group is assigned. You can use a wildcard in this field.

Build Using Map

Indicates whether the environment map is searched when building a list of elements. Values are:

Y—Search the map for all occurrences of the element.

N—Do not search the map. This is the default.

Display System/Subsys List

Indicates whether the system and subsystem selection lists will appear when the system or subsystem fields do not contain explicit entries. Values are:

Y—Display the system and subsystem selection lists.

N—Do not display the system and subsystem selection lists. This is the default.

Note: If you select the CREATE option, the System and Subsystem Selection Lists are returned, even if this field is set to N. If you do not know the system or subsystem name, enter a partial name using a wildcard name mask, set the DISPLAY SYS/SBS List option to Y, and then press Enter to view the System Selection List or Subsystem Selection List panel.

5. Leave the OPTIONs field blank and press Enter.

The panel that opens depends on the information you entered on the Display Element/Component Lists panel.

- If you fully qualified all search criteria, or wildcarded the stage ID, but there is only one element to match, the information for that element is displayed.
- If you left the ELEMENT field blank and set BUILD USING MAP=Y, the Element Name Prompt panel opens, provided your CA Endeavor SCM administrator set the element name prompt option for your site on the ENCOPTBL.
- If you wildcarded the ELEMENT field and set the DISPLAY LIST option to N, the Confirmation panel opens. The information displayed on this panel depends on how you set the BUILD USING MAP option:
 - If BUILD USING MAP=Y, the panel displays the number of elements selected from all environments.
 - If BUILD USING MAP=N, the panel displays the number of elements selected from the current environment.

Note: Use DISPLAY LIST = N with caution, especially in conjunction with BUILD USING MAP = Y. When you press Enter at a Confirmation panel, you will have to view all the elements that have been selected.

Press Enter to view the requested display for the number of elements indicated on the Confirmation panel.

- If you wildcarded the ELEMENT field and the DISPLAY LIST option is set to Y, the Element Selection List opens. The information displayed on this panel depends on how you set the BUILD USING MAP option.
 - If BUILD USING MAP=Y, the list displays all elements in all map environments that match your search criteria.
 - If BUILD USING MAP=N, the list displays all elements in the current environment that match your search criteria.

To view the Element Selection List, use one of the following methods.

- Scroll right to view more information.
- Scroll left (or right twice) to view all the information for each element on one panel, with two lines per element.

Note: You can scroll the panels by entering LEFT or RIGHT in the command line or by using the corresponding pfkeys, which are usually PF10 and PF11.

To view component list information for CA Endevor SCM ACM, enter **SX**, **BX**, **CX**, or **HX**.

Note: If you select multiple elements, press End to view the next panel of information.

- To view a summary of changes, enter **S** to the left of an element and press Enter to view a summary of change history for an element. You can then view additional information for one or more of the levels by entering a value to the left of the level and pressing Enter. (Enter **B** for the Element Browse panel, enter **C** for the Element Changes panel, or enter **H** for the Element History panel).
- To view element information from the Master Control File (MCF), enter **M** to the left of an element and press Enter to view Master Control File (MCF) information for an element.
- To view all statements in the current level, enter **B** to the left of an element and press Enter to view all of the statements in the element level and the level at which each statement was inserted. By default, the current level is displayed. To view previous levels, navigate to the Summary of Levels panel, enter **B** to the left of each level you want to view, and press Enter.
- To view all changes (insertions and deletions), enter **C** to the left of an element and press Enter to view all inserts and deletions made to the element between the specified level and its immediate predecessor. By default, the current level is displayed. To view the changes from previous levels, navigate to the Summary of Levels panel, enter **C** to the left of each level you want to view, and press Enter.

- To view all statements in all levels, enter **H** to the left of an element and press Enter to view all statements in all levels of the element, from the base level through the current level. You can view the level at which each insertion and deletion occurred. By default, the current level is displayed. To view the history of previous levels, navigate to the Summary of Levels panel, enter **H** to the left of each level you want to view, and press Enter.
- To view an element's output library listing, enter the List Listing option **LL** to the left of an element and press Enter.

Note: S, B, C, and H are invalid display options for sourceless elements. For elements of log delta format, options H and C are not valid, and option B can only be used on the current (last) level of the element. For elements of image delta format, H and C are not valid.

View Footprint Information

You can view *footprints* that are stored in PDS, CA Librarian, or CA Panvalet libraries. Footprints are used in the product to keep the source synchronized with executables.

To view footprint information

1. Start the product using the instructions provided by your site administrator.

The Primary Options panel appears.

2. Enter **1 (Display)** and press Enter.

The Display Options menu appears.

3. Enter **2 (Footprint)** and press Enter.

The Footprint Display panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. Select from the following options:

- To view a list of members, specify the range of members you want listed using the MEMBER and THRU MEMBER options, leave the Option field blank, and press Enter to view a Library Selection list of members in the specified library, including the footprint information for each non-load member (if available).
- To view a list of CSECTs, specify the library and member in which the load module is stored, enter **I** in the Option field (Footprint Display panel or Library Selection list), and press Enter to view a list of CSECTs for a specific load member, the footprint information for each CSECT, and the footprint for the associated load module.

- To view a member in an output or listing library, specify the library and member you want to view, enter **L** in the Option field (Footprint Display panel or Library Selection list), and press Enter to view the content of a specific non-load library member that is footprinted (including the source output library, generate processor output library, or move processor output library).
 - To view element information for members, CSECTs, or load modules, enter values in the FROM ISPF LIBRARY and OTHER PARTITIONED DATA SET options and press Enter to view element information for members, CSECTs, and load modules that are footprinted.
5. Review the footprint information.

View Site Definitions

You can view the *site definitions* that your site administrator set up during installation.

Note: All site information that you can view is maintained in your current Defaults Table (C1DEFLT5). For more information about this table, contact your site administrator.

To view site definitions

1. Start the product using the instructions provided by your site administrator.
The Primary Options panel appears.
2. Enter **1 (Display)** and press Enter.
The Display Options menu appears.
3. Enter **3 (Site)** and press Enter.
The Site Information from the C1DEFLT5 table appears.
Note: For more information about a field or option on the panel, press PF1.
4. Review the site information.

View Stage Information

You can view information about the two *stages* within your current environment that your site administrator set up during installation.

To view stage information

1. Start the product using the instructions provided by your site administrator.
The Primary Options panel appears.
2. Enter **1 (Display)** and press Enter.
The Display Options menu appears.

3. Enter **4 (Stage)** and press Enter.

The Stage Information panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. Review the stage information for the current environment.

Note: To review the stages for a different environment, enter the environment name in the CURRENT ENV option and press Enter.

View System Definitions

You can view information about the different *systems* that your site administrator has set up.

To view system definitions

1. Start the product using the instructions provided by your site administrator.

The Primary Options panel appears.

2. Enter **1 (Display)** and press Enter.

The Display Options menu appears.

3. Enter **5 (System)** and press Enter.

The System Display panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. Enter the information associated with the system you want to view (environment and system) and press Enter.

Note: If you do not know the system name, leave the SYSTEM option blank, or enter a partial name using a name mask, and press Enter to view the System Selection List panel to view all systems for the current environment.

5. (Optional) If the System Selection List panel appears, enter **S** to the left of each system you want to view and press Enter.

The System Definition panel appears.

6. Review the system definition for the current environment.

Note: If you selected multiple system definitions to view, press Enter to view the next system definition.

View Subsystem Definitions

You can view information about the different *subsystems* that your site administrator has set up.

To view subsystem definitions

1. Start the product using the instructions provided by your site administrator.

The Primary Options panel appears.

2. Enter **1 (Display)** and press Enter.

The Display Options menu appears.

3. Enter **6 (Subsystem)** and press Enter.

The Subsystem Display panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. Enter the information associated with the subsystem you want to view (environment, system, and subsystem) and press Enter.

Note: If you do not know the system or subsystem name, leave the SYSTEM or SUBSYSTEM options blank, or enter a partial name using a name mask, and press Enter to view the System Selection List or Subsystem Selection List panel to view all systems or subsystems for the current environment.

5. (Optional) If the System Selection List or Subsystem Selection List panel appears, enter **S** to the left of each system or subsystem you want to view and press Enter.

The Subsystem Definition panel appears.

6. Review the subsystem definition for the current environment.

Note: If you selected multiple subsystem definitions to view, press Enter to view the next subsystem definition.

View Type Definitions

You can view information about the different *types* that your site administrator has set up for each *stage* within a *system*.

To view type definitions

1. Start the product using the instructions provided by your site administrator.

The Primary Options panel appears.

2. Enter **1 (Display)** and press Enter.

The Display Options menu appears.

3. Enter **7 (Type)** and press Enter.

The Type Display panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. Enter the information associated with the type you want to view (environment, system, type, and stage) and press Enter.

Note: If you do not know the system or type name, leave the SYSTEM or TYPE options blank, or enter a partial name using a name mask, and press Enter to view the System Selection List or Type Selection List panel to view all systems or types for the current environment.

5. (Optional) If the Type Selection List panel appears, enter **S** to the left of each type you want to view and press Enter.

The Type Definition panel appears.

6. Review the type definition for the current environment, stage, and system. If you selected multiple type definitions to view, press Enter to view the next type definition.

Note: If you are using CA Panvalet or CA Librarian to store elements, your site administrator is responsible for defining the appropriate source language that you can view in the PV/LB LANG option on the Type Definition panel.

View Processor Group Definitions

You can view information about the different *processor groups* that your site administrator has set up.

To view processor group definitions

1. Start the product using the instructions provided by your site administrator.

The Primary Options panel appears.

2. Enter **1 (Display)** and press Enter.

The Display Options menu appears.

3. Enter **8 (Processor Group)** and press Enter.

The Processor Group Display panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. Enter the information associated with the processor group you want to view (environment, system, type, stage, and group) and press Enter.

Note: If you do not know the system, type, or processor group name, leave the SYSTEM, TYPE, or GROUP options blank, or enter a partial name using a name mask, and press Enter to view the System Selection List, Type Selection List, or Processor Group Selection List panel to view all systems, types, or processor groups for the current environment.

5. (Optional) If the Processor Group Selection List panel appears, enter **S** to the left of each processor group you want to view and press Enter. If you select a processor group definition containing user-defined *symbolics*, enter **S** to the left of the processor group and press Enter to display the Processor Group Symbolics panel. From this panel, you can view and override user-defined symbolics for the selected processor group.

The Processor Group Definition panel appears. This panel (for type Process) does not provide information about the next environment, the next processor group, or selection options for the OUTPUT MANAGEMENT INFORMATION options because the system definition only allows for two processor load libraries to be defined.

Note: To view the JCL for the processors in a processor group, enter **L** to the left of each processor name and press Enter to display the appropriate Processor JCL panels.

6. Review the processor group definition for the current environment, stage, system, and type.

Note: If you selected multiple processor group definitions to view, press Enter to view the next processor group definition.

View Approver Group Definitions

You can view information about the different *approver groups* that your site administrator has set up within each *environment*.

To view approver group definitions

1. Start the product using the instructions provided by your site administrator.

The Primary Options panel appears.

2. Enter **1 (Display)** and press Enter.

The Display Options menu appears.

3. Enter **9 (Approver Group)** and press Enter.

The Approver Group Display panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. Enter the information associated with the approver group you want to view (environment and approver group) and press Enter.

Note: If you do not know the approver group name, leave the APPROVER GROUP option blank, or enter a partial name using a name mask, and press Enter to view the Approver Group Selection List panel to view all standard and emergency approver groups for the current environment.

5. (Optional) If the Approver Group Selection List panel appears, enter **S** to the left of each approver group you want to view and press Enter.

The Approver Group Definition panel appears.

6. Review the approver group definition for the current environment.

Note: If you selected multiple approver group definitions to view, press Enter to view the next approver group definition.

View Approver Group Relationship Definitions

You can view information about the different *approver group* relationships that your site administrator has set up within each *environment*.

To view approver group relationship definitions

1. Start the product using the instructions provided by your site administrator.

The Primary Options panel appears.

2. Enter **1 (Display)** and press Enter.

The Display Options menu appears.

3. Enter **A (Relate Group)** and press Enter.

The Approver Group Relation Display panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. Enter the information to identify the inventory area for which you want to view related approver groups (environment, system, subsystem, type, stage number, and approver type) and press Enter. If you do not know the name of the inventory area, leave the SYSTEM, SUBSYSTEM, TYPE, or STAGE NUMBER options blank and press Enter to display a list.

Note: You cannot use a *name mask* when entering information on the Approver Group Relation Display panel. An asterisk (*), normally used to indicate a name mask, represents actual data. For example, if you enter an asterisk as the subsystem name, the product records (and displays) that value as the actual name. The Approver Group Relationship List then displays only those relationships with "*" as the value for the SUBSYSTEM option.

5. (Optional) If the Approver Group Relationship List panel appears, enter **S** to the left of each approver group relationship you want to view and press Enter.

The Approver Group Relationship panel appears.

6. Review the approver group relationship for the current environment.

Note: If you selected multiple approver group relationships to view, press Enter to view the next approver group relationship.

View Environment Information

You can view the *environment information* that your site administrator set up during installation.

Note: All environment information that you can view is maintained in your current Defaults Table (C1DEFLT5). For more information about this table, contact your site administrator.

To view environment information

1. Start the product using the instructions provided by your site administrator.

The Primary Options panel appears.

2. Enter **1 (Display)** and press Enter.

The Display Options menu appears.

3. Enter **E (Environment)** and press Enter.

The Environment Information panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. Review the environment information.

Note: To review the information for a different environment, enter the environment name in the Current Environment option and press Enter.

View Site Symbolics

You can view the *site symbolics* that your site administrator set up during installation.

To view site symbolics

1. Start the product using the instructions provided by your site administrator.

The Primary Options panel appears.

2. Enter **1 (Display)** and press Enter.

The Display Options menu appears.

3. Enter **S (Site Symbols)** and press Enter.

The Site Symbols Information panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. Review the site symbolics.

View Global Type Processing Sequence

If your site administrator has enabled *global type sequencing*, you can view the *processing sequence* for types defined at your site.

To view the processing sequence for types

1. Start the product using the instructions provided by your site administrator.
The Primary Options panel appears.
2. Enter **1 (Display)** and press Enter.
The Display Options menu appears.
3. Enter **T (Type Sequence)** and press Enter.
The Type Processing Sequence panel appears.

Note: For more information about a field or option on the panel, press PF1.

4. Review the processing sequence.

Appendix A: Selecting the Correct Processor Group

This section contains the following topics:

- [Element Actions and Processor Groups](#) (see page 91)
- [Add Elements Action and Processor Groups](#) (see page 96)
- [Add/Update Elements Actions and Processor Groups](#) (see page 97)
- [Delete Elements Action and Processor Groups](#) (see page 98)
- [Generate Elements Action and Processor Groups](#) (see page 99)
- [Generate Elements with Copyback and Processor Groups](#) (see page 101)
- [Restore Elements Action and Processor Groups](#) (see page 102)
- [Move Elements Action \(Using a Move Processor\) and Processor Groups](#) (see page 103)
- [Move Elements Action \(Using a Generate Processor\) and Processor Groups](#) (see page 105)
- [Transfer Elements Action \(Using a Move Processor\) and Processor Groups](#) (see page 108)
- [Transfer Action \(Using a Generate Processor\) and Processor Groups](#) (see page 111)

Element Actions and Processor Groups

When you perform a development task, you perform an action on an *element*. For example, to manage a member from an external data set, you use the Add Elements action. Each element is associated with a *processor group* through the *element type*. The following element actions determine which processor group is selected:

- Add Elements
- Add/Update Elements
- Delete Elements
- Update Elements
- Generate Elements
- Generate Elements (with Copyback)
- Restore Elements
- Move Elements (using the move processor)

- Move Elements (using the generate processor)
- Transfer Elements (using the move processor)
- Transfer Elements (using the generate processor)

When you use an element action, the processor group from which a processor is run may change, based on the options you select to execute the action (for example, the element location, mapping, processor group overrides, and so forth). As a result, you may see unexpected results and not understand why a specific processor group is being used to perform the action.

This appendix provides information to help you understand how the processor group to use for an element action is determined. For each action, you will see the “expected” results for a specific set of conditions. This information will help you determine whether the appropriate processor group name and definition is associated with the elements you are working with. Compare the information in this appendix to your results when the action is executed.

If you find results that are different from this information, double-check your element action criteria to be sure that the conditions for the action match. If the conditions are different, the expected results may not occur. If, however, your element action criteria match this information and you do not get the same results, contact Technical Support at <http://ca.com/support> for assistance.

More information:

[Variables in Processor Group Information](#) (see page 94)

[Processor Group Rules](#) (see page 95)

[Processor Group Example Criteria](#) (see page 95)

Processor Group Information

For each element action in this appendix, the processor group information is organized as follows:

Case

Provides a cross-reference to examples that explain the information for the element action. Examples are not provided for every case. Use the examples as a guideline to help you determine which processor is executed and which label is assigned.

Conditions

Displays the conditions under which you execute the action. You may see one of the following entries in this column:

Element exists at target

Displays Y (Yes) or N (No) to indicate whether the element you are working with exists at the target location.

Element exists up map

Displays Y (Yes) or N (No) to indicate whether the element you are working with exists up the map.

Processor Group override (client)

Displays Y (Yes) or N (No) to indicate whether you have manually overridden the processor group name at this time, for this element, for this action.

Processor Group mapped (system-defined)

Displays Y (Yes) or N (No) to indicate whether your site administrator has defined a different processor group for this element at the next map location.

Processor Group change

Displays Y (Yes) or N (No) to indicate whether the processor group at the target location is changing. A change can occur only if the processor group has been overridden or mapped, and the element exists at the target.

Processor

Displays the processor group that you should expect to be used as a result of the element action under the specific conditions. You may see one of the following entries in this column:

Target (T)/Source (S)--Processor Type

Indicates the type of processor executed as a result of the action (Generate, Move, or Delete processor) and whether the processor was at the target location (T) or the source location (S). This column has the following variable format:

S/T)-processor type

Processor Group Name taken from

Indicates the location used to create the processor group name for this action, for example, the source element or the target element.

Processor Group Definition taken from

Indicates the located used to create the processor group definition for this action, for example, the target or source.

Processor Group Name assigned

Indicates where the product got the processor group name that is assigned to the element when the action has completed, for example, the fetched element or the target element.

Variables in Processor Group Information

For each element action in this appendix, variables are used to indicate the location from which information is taken. These variables are defined as follows:

PG chg

Indicates that the processor group has changed, based on the conditions for the action (override or mapping).

Fetched element

The element copied back to the TO location (for Add and Transfer actions), and to the FROM location of a Generate with Copyback action.

Target default

The default processor group name for the element type at the target location.

Override

The processor group specified using the PROCESSOR GROUP OVERRIDE option. Processor group overrides are performed explicitly, on an action-by-action basis.

Target element (Target)

The element at the target (the “TO” location) of the action, or the element that is being updated, generated (without copyback), or deleted.

Source element (Source)

The element at the source (the “FROM” location) of the action.

Map

The processor group specified using the NEXT PROCESSOR GROUP option, which indicates the name of the processor group at the next map location. NEXT PROCESSOR GROUP is assigned by your site administrator and cannot be changed for a specific action. If no processor group is assigned up the map, the product uses the source element processor group.

Processor Group Rules

Consider the following information when using processor groups:

- A processor group change may occur only when the element exists at the target location.
- You cannot override a processor group on a Move action.
- For Move and Transfer actions, the source processor group always determines the type of processor (generate or move) to be executed.

Processor Group Example Criteria

For each element action in this appendix, you will see one or more examples. An example is not provided for every case.

The following table contains the criteria that is used for all examples in this appendix:

	Environment 1 Stage 1	Environment 1 Stage 2	Environment 2 Stage 3	Environment 2 Stage 4
Type	COBOL	COBOL	COBOL	COBOL
Processor Group Name	LOPGA	LOPGA	LOPGA	LOPGA
Generate Processor	SIMPLGA	SIMPLGA	SIMPLGA	SIMPLGA
Delete Processor	SIMPLDA	SIMPLDA	SIMPLDA	SIMPLDA
Move Processor	SIMPLMA	SIMPLMA	SIMPLMA	SIMPLMA
Processor Group Name	LOPGB	LOPGB	LOPGB	LOPGB
Generate Processor	SIMPLGB	SIMPLGB	SIMPLGB	SIMPLGB
Delete Processor	SIMPLDB	SIMPLDB	SIMPLDB	SIMPLDB
Move Processor	SIMPLMB	SIMPLMB	SIMPLMB	SIMPLMB
Processor Group Name	LOPGC	LOPGC	LOPGC	LOPGC
Generate Processor	SIMPLGC	SIMPLGC	SIMPLGC	SIMPLGC
Delete Processor	SIMPLDC	SIMPLDC	SIMPLDC	SIMPLDC
Move Processor	SIMPLMC	SIMPLMC	SIMPLMC	SIMPLMC

Add Elements Action and Processor Groups

The following table describes how the Add Elements action determines which processor group is used:

Case	Conditions	Processor
1	<ul style="list-style-type: none"> ■ Element exists up map: Yes ■ Processor group override (client): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group name taken from: fetched element ■ Processor Group taken from: target ■ Processor Group Name assigned: fetched element
2	<ul style="list-style-type: none"> ■ Element exists up map: No ■ Processor Group override (client): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group name taken from: target default ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: target default
3	<ul style="list-style-type: none"> ■ Element exists up map: Yes ■ Processor Group override (client): Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor Type: T-Generate ■ Processor Group name taken from: override ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: override
4	<ul style="list-style-type: none"> ■ Element exists up map: No ■ Processor Group override (client): Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor Type: T-Generate ■ Processor Group Name taken from: override ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: override

Add Elements Action, Case 1 Example

Example: Add element FINAPP01

In this example, you add element FINAPP01 to type COBOL. The element exists up the map at Stage 3, and is labeled with processor group LOPGC. You do not provide a processor group override for the action.

As a result, the SIMPLGC processor from the processor group LOPGC of Stage 1 is executed and the LOPGC label is assigned.

Add/Update Elements Actions and Processor Groups

The following table describes how the Add/Update Elements action determines which processor group is used:

Case	Conditions	Processor
1	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: Yes ■ Processor Group override (client): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group name taken from: target element ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: target element
2	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: No ■ Processor Group override (client): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group name taken from: target element ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: target element
3	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: Yes ■ Processor Group override (client): Yes ■ Processor Group change: Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Delete (PG chg) T-Generate ■ Processor Group name taken from: target element override ■ Processor Group Definition taken from: target element ■ Processor Group Name assigned: override
4	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: No ■ Processor Group override (client): Yes ■ Processor Group change: Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Delete (PG chg) T-Generate ■ Processor Group name taken from: target element ■ Processor Group Definition taken from: target element ■ Processor Group Name assigned: override

More information:

[Add/Update Elements Action, Case 3 Example](#) (see page 98)

Add/Update Elements Action, Case 1 Example

Example: Add element FINAPP01

In this example, you add element FINAPP01 to type COBOL. The element exists at the target and up the map, at Stage 4, labeled with processor group LOPGB. You do not provide a processor group override for the action.

As a result, the SIMPLGB processor from Stage 1 is executed and the LOPGC label is assigned.

Add/Update Elements Action, Case 3 Example

Example: Add element FINAPP01 (processor group override)

In this example, you add element FINAPP01 to type COBOL. The element exists at the target and up the map, labeled with processor group LOPGB. You provide a processor group override, LOPGA, which causes the processor group name to change.

As a result, the SIMPLDB processor from Stage 1 and the SIMPLGA processor from Stage 1 are executed, and the LOPGA label is assigned.

Delete Elements Action and Processor Groups

The following table describes how the Delete Elements action determines which processor group is used:

Case	Conditions	Processor
1	<ul style="list-style-type: none">Element exists at target: Yes	<ul style="list-style-type: none">Target (T)/Source (S) Processor Type: T-DeleteProcessor Group name taken from: target elementProcessor Group Definition taken from: targetProcessor Group Name assigned: target element

Delete Elements Action, Case 1 Example

Example: Delete element FINAPP02

In this example, you delete element FINAPP02 from Stage 2. The element exists at the target only, and is labeled with processor group LOPGB. As a result, the SIMPLDB processor from Stage 2 is executed and no label is assigned, since the element is deleted.

Generate Elements Action and Processor Groups

The following table describes how the Generate Elements action determines which processor group is used:

Case	Conditions	Processor
1	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: Yes ■ Processor Group override (client): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group name taken from: target element ■ Processor Group taken from: target ■ Processor Group Name assigned: target element
2	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: No ■ Processor Group override (client): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group Name taken from: target element ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: target element

Case	Conditions	Processor
	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: Yes ■ Processor Group override (client): Yes ■ Processor Group change: Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Delete (PG chg) T-Generate ■ Processor Group name taken from: target element override ■ Processor Group Definition taken from: target element ■ Processor Group Name assigned: override
4	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: No ■ Processor Group override (client): Yes ■ Processor Group change: Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Delete (PG chg) T-Generate ■ Processor Group Name taken from: target element override ■ Processor Group Definition taken from: target element ■ Processor Group Name assigned: override

More information:

[Generate Elements Action, Case 4 Example](#) (see page 100)

Generate Elements Action, Case 1 Example

Example: Generate element FINAPP01

In this example, you generate element FINAPP01 at Stage 2 where it is labeled with processor group LOPGB. You do not provide a processor group override for the action.

As a result, the SIMPLGB processor group from Stage 2 is executed and the LOPGB label is assigned.

Generate Elements Action, Case 4 Example

Example: Generate element FINAPP01 (processor group override)

In this example, you generate element FINAPP01 at Stage 2 where it is labeled with processor group LOPGB. You provide a processor group override for the action, LOPGC, which causes the processor group name to change.

As a result, the SIMPLDB processor from Stage 2 and the SIMPLGC processor from Stage 2 are executed, and the LOPGC label is assigned.

Generate Elements with Copyback and Processor Groups

The following table describes how the Generate Elements with Copyback action determines which processor group is used:

Case	Conditions	Processor
1	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: Yes ■ Processor Group override (client): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group name taken from: fetched element ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: fetched element
2	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: Yes ■ Processor Group override (client): Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group Name taken from: override ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: override

Generate Elements with Copyback Action, Case 1 Example

Example: Generate element FINAPP01

In this example, you generate element FINAPP01 at Stage 1 with copyback and you do not provide a processor group override for the action. The element exists at Stage 4 and is assigned to processor group LOPGA.

As a result, the SIMPLGA processor from Stage 1 is executed and the LOPGA label is assigned.

Restore Elements Action and Processor Groups

The following table describes how the Restore Elements action determines which processor group is used:

Case	Conditions	Processor
1	<ul style="list-style-type: none"> ■ Element exists up map: Yes ■ Processor Group override: No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group Name taken from: source element ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: source element
2	<ul style="list-style-type: none"> ■ Element exists up map: No ■ Processor Group override (client): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group Name taken from: source element ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: source element
3	<ul style="list-style-type: none"> ■ Element exists up map: Yes ■ Processor Group override (client): Yes ■ Processor Group change: Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group Name taken from: override ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: override
4	<ul style="list-style-type: none"> ■ Element exists up map: No ■ Processor Group override (client): Yes ■ Processor Group change: Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group Name taken from: override ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: override

Restore Elements Action, Case 1 Example

Example: Restore element FINAPP04

In this example, you restore element FINAPP04 to Stage 1. The element exists up the map, at Stage 3, with the processor group LOPGB. You do not provide a processor group override for the action.

As a result, the SIMPLGB processor from Stage 1 is executed and the LOPGB label is assigned.

Move Elements Action (Using a Move Processor) and Processor Groups

The following table describes how the Move Elements action (using a move processor) determines which processor group is used:

Case	Conditions	Processor
1	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: Yes ■ Processor Group mapped (system-defined): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move S-Delete ■ Processor Group Name taken from: source element source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: source element
2	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: No ■ Processor Group mapped (system-defined): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move S-Delete ■ Processor Group Name taken from: source element source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: source element
3	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: Yes ■ Processor Group mapped (system-defined): No ■ Processor Group change: No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move S-Delete ■ Processor Group Name taken from: source element source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: source element
4	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: No ■ Processor Group mapped (system-defined): No ■ Processor Group change: No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move S-Delete ■ Processor Group Name taken from: source element source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: source element

Case	Conditions	Processor
5	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: Yes ■ Processor Group mapped (system-defined): Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move S-Delete ■ Processor Group Name taken from: source element source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: map
6	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: No ■ Processor Group mapped (system-defined): Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move S-Delete ■ Processor Group Name taken from: source element source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: map
7	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: Yes ■ Processor Group mapped (system-defined): Yes ■ Processor Group change: Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Delete (PG chg) S-Move S-Delete ■ Processor Group Name taken from: target element source element source element ■ Processor Group Definition taken from: target source ■ Processor Group Name assigned: map
8	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: No ■ Processor Group mapped (system-defined): Yes ■ Processor Group change: No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move S-Delete ■ Processor Group Name taken from: source element source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: map

More information:

[Move Elements Action \(Using a Move Processor\), Case 7 Example](#) (see page 105)

Move Elements Action (Using a Move Processor), Case 1 Example

Example: Move element FINAPP01 from Stage 1 to Stage 2

In this example, you move element FINAPP01 from Stage 1 to Stage 2. In Stage 1, the element is labeled with processor group LOPGC. The element does not exist at the target, but exists up the map, at Stage 4, with processor group LOPGC. A processor group name is not mapped at the target (next) location.

As a result, the SIMPLMC processor from Stage 1 and the SIMPLDC processor from Stage 1 are executed, and the LOPGC label is assigned.

Move Elements Action (Using a Move Processor), Case 7 Example

Example: Move element FINAPP01 from Stage 2 to Stage 3

In this example, you move element FINAPP01 from Stage 2 to Stage 3. In Stage 2, the element is labeled with processor group LOPGA. The element exists at the target, with processor group LOPGB, and up the map, at Stage 4. A processor group name is mapped at the target (next) location of LOPGC. As a result, the processor group name changes.

As a result, the SIMPLDB processor from Stage 3, the SIMPLMA processor from Stage 2, and the SIMPLDA processor from Stage 2 are executed, and the LOPGC label is assigned.

Move Elements Action (Using a Generate Processor) and Processor Groups

The following table describes how the Move Elements action (using a generate processor) determines which processor group is used:

Case	Conditions	Processor
1	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: Yes ■ Processor Group mapped (system-defined): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate S-Delete ■ Processor Group Name taken from: source element ■ Processor Group Definition taken from: target source ■ Processor Group Name assigned: source element

Case	Conditions	Processor
2	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: No ■ Processor Group mapped (system-defined): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate S-Delete ■ Processor Group Name taken from: source element ■ Processor Group Definition taken from: target source ■ Processor Group Name assigned: source element
3	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: Yes ■ Processor Group mapped (system-defined): No ■ Processor Group change: No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate S-Delete ■ Processor Group Name taken from: source element ■ Processor Group Definition taken from: target source ■ Processor Group Name assigned: source element
4	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: No ■ Processor Group mapped (system-defined): No ■ Processor Group change: No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate S-Delete ■ Processor Group Name taken from: source element ■ Processor Group Definition taken from: target source ■ Processor Group Name assigned: source element
5	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: Yes ■ Processor Group mapped (system-defined): Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate S-Delete ■ Processor Group Name taken from: map source element ■ Processor Group Definition taken from: target source ■ Processor Group Name assigned: map
6	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: No ■ Processor Group mapped (system-defined): Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate S-Delete ■ Processor Group Name taken from: map source element ■ Processor Group Definition taken from: target source ■ Processor Group Name assigned: map

Case	Conditions	Processor
7	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: Yes ■ Processor Group mapped (system-defined): Yes ■ Processor Group change: Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Delete (PG chg) T-Generate S-Delete ■ Processor Group Name taken from: target element map source element ■ Processor Group Definition taken from: target source ■ Processor Group Name assigned: map
8	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: No ■ Processor Group mapped (system-defined): Yes ■ Processor Group change: No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate S-Delete ■ Processor Group Name taken from: map source element ■ Processor Group Definition taken from: target source ■ Processor Group Name assigned: map

More information:

[Move Elements Action \(Using a Generate Processor\), Case 5 Example](#) (see page 107)

Move Elements Action (Using a Generate Processor), Case 3 Example

Example: Move element FINAPP01 from Stage 1 to Stage 2

In this example, you move element FINAPP01 from Stage 1 to Stage 2. The element exists at the target, labeled with processor group LOPGB, and up the map, at Stage 4. A processor group name is not mapped at the target (next) location.

As a result, the SIMPLGB processor from Stage 2 and the SIMPLDB processor from Stage 1 are executed, and the LOPGB label is assigned.

Move Elements Action (Using a Generate Processor), Case 5 Example

Example: Move element FINAPP01 from Stage 2 to Stage 3

In this example, you move element FINAPP01 from Stage 2 to Stage 3. The element at Stage 2 is labeled with processor group LOPGB. The element does not exist at the target location but exists up the map at Stage 4. A processor group name is mapped at the target (next) location--LOPGC.

As a result, the SIMPLGC processor from Stage 3 and the SIMPLDB processor from Stage 2 are executed, and the LOPGC label is assigned.

Transfer Elements Action (Using a Move Processor) and Processor Groups

The following table describes how the Transfer Elements action (using a move processor) determines which processor group is used.

In the following table, the first eight rows illustrate that similar sets of conditions can produce different results. This can happen because of a change in the inventory component, such as type or system. One set of conditions (labeled with the letter A) reflects a transfer in which the source and target inventory components are the same. A second set of conditions (labeled with the letter B) reflects a transfer in which the source and inventory components are not the same. Compare lines A1 and B1, A2 and B2, and so forth. Although the conditions for executing the actions are similar, the results will be different.

The last four lines of the table present unique sets of conditions for the Transfer action:

Case	Conditions	Processor
1(A1)	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: Yes ■ Processor Group override (client): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move ■ Processor Group Name taken from: source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: source element
1(A2)	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: No ■ Processor Group override (client): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move ■ Processor Group Name taken from: source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: source element
2(B1)	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: Yes ■ Processor Group override (client): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move ■ Processor Group Name taken from: source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: target default
2(B2)	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: No ■ Processor Group mapped (system-defined): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move ■ Processor Group Name taken from: source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: target default

Case	Conditions	Processor
3(A3)	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: Yes ■ Processor Group mapped (system-defined): No ■ Processor Group Change: No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move ■ Processor Group Name taken from: target element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: target element
3(A4)	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: No ■ Processor Group override (client): No ■ Processor Group change: No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move ■ Processor Group Name taken from: target element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: target element
4(B3)	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: Yes ■ Processor Group override (client): No ■ Processor Group change: No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move ■ Processor Group Name taken from: source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: target element
4(B4)	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: No ■ Processor Group override (client): No ■ Processor Group change: No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move ■ Processor Group Name taken from: source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: target element
5	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: Yes ■ Processor Group mapped (system-defined): Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move ■ Processor Group Name taken from: source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: override
6	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: No ■ Processor Group mapped (system-defined): Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move ■ Processor Group Name taken from: source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: override
7	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: Yes ■ Processor Group mapped (system-defined): Yes ■ Processor Group Change: Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Delete (PG chg) S-Move ■ Processor Group Name taken from: target element source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: override

Case	Conditions	Processor
8	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: No ■ Processor Group mapped (system-defined): Yes ■ Processor Group change: Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Delete (PG chg) S-Move ■ Processor Group Name taken from: target element source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: override

More information:

[Transfer Elements Action \(Using a Move Processor\), Case 7 Example](#) (see page 110)

Transfer Elements Action (Using a Move Processor), Case 1 (A1) Example

Example: Transfer element FINAPP01 from Stage 1 to Stage 3

In this example, you transfer element FINAPP01 from Stage 1 to Stage 3. In Stage 1, the element is labeled with processor group LOPGA. The element does not exist at the target, but exists up the map, at Stage 4, labeled with the processor group LOPGA. You do not provide a processor group override for the action.

As a result, the SIMPLMA processor from Stage 1 is executed and the LOPGA label is assigned.

Transfer Elements Action (Using a Move Processor), Case 7 Example

Example: Transfer element FINAPP01 from Stage 1 to Stage 3 (processor group override)

In this example, you transfer element FINAPP01 from Stage 1 to Stage 3. In Stage 1, the element is labeled with processor group LOPGA. The element exists at the target (Stage 3) and is labeled with the processor group LOPGA. You provide a processor group override for the action--LOPGC--which results in the processor group name changing.

As a result, the SIMPLDA processor from Stage 3 and the SIMPLMA processor from Stage 1 are executed, and the LOGPC label is assigned.

Transfer Action (Using a Generate Processor) and Processor Groups

The following table describes how the Transfer Elements action (using a generate processor) determines which processor group is used.

In the following table, the first eight rows illustrate that similar sets of conditions can produce different results. This can happen because of a change in the inventory component, such as type or system. One set of conditions (labeled with the letter A) reflects a transfer in which the source and target inventory components are the same. A second set of conditions (labeled with the letter B) reflects a transfer in which the source and inventory components are not the same. Compare lines A1 and B1, A2 and B2, and so forth. Although the conditions for executing the actions are similar, the results will be different.

The last four lines of the table present unique sets of conditions for the Transfer action:

Case	Conditions	Processor
1(A1)	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: Yes ■ Processor Group override (client): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group Name taken from: source element ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: source element
1(A2)	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: No ■ Processor Group override (client): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group Name taken from: source element ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: source element
2(B1)	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: Yes ■ Processor Group override (client): No 	<ul style="list-style-type: none"> ■ Target (T) /Source (S) Processor type: T-Generate ■ Processor Group Name taken from: target default ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: source element
2(B2)	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: No ■ Processor Group override (client): No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group Name taken from: target default ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: target default

Case	Conditions	Processor
3(A3)	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: Yes ■ Processor Group override (client): No ■ Processor Group change: No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group Name taken from: target element ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: target element
3(A4)	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: No ■ Processor Group override (client): No ■ Processor Group change: No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group Name taken from: target element ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: target element
4(B3)	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: Yes ■ Processor Group override (client): No ■ Processor Group change: No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group Name taken from: target element ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: target element
4(B4)	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: No ■ Processor Group override (client): No ■ Processor Group change: No 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group Name taken from: target element ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: target element
5	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: Yes ■ Processor Group override (client): Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group Name taken from: override ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: override
6	<ul style="list-style-type: none"> ■ Element exists at target: No ■ Element exists up map: No ■ Processor Group override (client): Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Generate ■ Processor Group Name taken from: override ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: override
7	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: Yes ■ Processor Group override (client): Yes ■ Processor Group change: Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Delete (PG chg) T-Generate ■ Processor Group Name taken from: target element override ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: override

Case	Conditions	Processor
8	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: No ■ Processor Group override (client): Yes ■ Processor Group change: Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: T-Delete (PG chg) T-Generate ■ Processor Group Name taken from: target element override ■ Processor Group Definition taken from: target ■ Processor Group Name assigned: override
9(B1)	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: No ■ Processor Group override (client): No ■ Processor Group Change: Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move ■ Processor Group Name taken from: source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: target default
9(B2)	<ul style="list-style-type: none"> ■ Element exists at target: Yes ■ Element exists up map: Yes ■ Processor Group override (system-defined): No ■ Processor Group Change: Yes 	<ul style="list-style-type: none"> ■ Target (T)/Source (S) Processor type: S-Move ■ Processor Group Name taken from: source element ■ Processor Group Definition taken from: source ■ Processor Group Name assigned: target default ■ Sync required

More information:

[Transfer Elements Action \(Using a Generate Processor\), Case 8 Example](#) (see page 114)

Transfer Elements Action (Using a Generate Processor), Case 1 (A2) Example

Example: Transfer element FINAPP01 from Stage 1 to Stage 3

In this example, you transfer element FINAPP01 from Stage 1 to Stage 3. The element exists at Stage 1 with the processor group LOPGA. The element does not exist at the target or up the map. You do not provide a processor group override for the action.

As a result, the SIMPLGA processor from Stage 3 is executed and the LOPGA label is assigned.

Transfer Elements Action (Using a Generate Processor), Case 8 Example

Example: Transfer element FINAPP01 from Stage 1 to Stage 3 (processor group override)

In this example, you transfer element FINAPP01 from Stage 1 to Stage 3. The element exists at Stage 1 with processor group LOPGB. The element exists at the target, with processor group LOPGB, but does not exist up the map. You provide a processor group override for the action LOPGC which results in the processor group name changing.

As a result, the SIMPLDB processor from Stage 3 and the SIMPLGC processor from Stage 3 execute, and the LOPGC label is assigned.

Glossary

Access Security table

The *Access Security Table* is the CA Endeavor SCM table that defines the environments to which users have access. Each site has one Access Security Table.

action

An *action* is a command to manage elements.

Add action

The *Add action* places members from an external data set or USS directory under control of CA Endeavor SCM.

alternate ID

An *alternate ID* is a facility that protects the data sets controlled by CA Endeavor SCM (such as the Master Control Files, package data sets, and base and delta libraries) from access by users, while still allowing CA Endeavor SCM to have access. The alternate ID is defined by the RACFUID parameter in the CA Endeavor SCM Defaults Table.

append SCL

To *append SCL* is to add new SCL statements to the end of an existing data set, rather than overwriting the existing SCL. This option is available on the Batch Options Menu when building element action SCL in foreground for execution in batch, and it is also available on the Create/Modify Package panel when creating or modifying a package.

approval

An *approval* is an electronic signoff mechanism for packages. An approval may be required for a package before execution.

Approved

Approved is the status of a package after the required approvers have reviewed, signed off, and approved the package.

approver

An *approver* is a person authorized to sign off on a package, before execution. Signing off on a package involves reviewing the information in the package and approving or denying the package.

approver group

An *approver group* is a collection of one or more approvers. An approver group is defined within each environment and can be associated with inventory areas.

approver group relationship

An *approver group relationship* is established between an approver group and one or more inventory areas, authorizing members of the approver group to review (and then approve or deny) packages related to the inventory areas.

approver type

An *approver type* is the specification of the kinds of packages that an approver group can review. When the approver type is Standard, the approver group can review only standard packages related to its authorized inventory areas. When the approver type is Emergency, the approver group can review only emergency packages related to its authorized inventory areas.

Archive action

The *Archive action* writes an element and all related CA Endeavor SCM information to a sequential data set.

Autogen action option

The *Autogen action option*, when specified for an element, automatically generates the elements that use this component element. For example, specifying Autogen for an element of type COPYBOOK would automatically generate the COBOL PROGRAM elements that use the macro, which would in turn generate the appropriate LNK elements. An element that uses a component element is known as a *using* element. The using elements generated on behalf of Autogen are generated with the NOSOURCE option or the COPYBACK option, depending on the ENCOPTBL setting. Autogen is available in batch only for the Add, Update, and Generate actions and cannot be used in packages.

Automated Configuration Manager (ACM)

The *Automated Configuration Manager (ACM)* is an optional facility that allows you to monitor selected libraries and data sets and maintain a component list for each element in the monitored areas. The component list provides an audit trail of program-component information at the time of each compilation.

backin

The *backin* command restores the executable members of a package to the state they were in before the package was backed out. A backin reverses the backout process.

backout

The *backout* command returns the executable members of a package to the state they were in before package execution.

base level

The *base level* is the lowest level of an element within a particular stage when storing elements with forward deltas. This level represents the source for the element in that stage. If an element exists in both stages, there is a base level in each stage. When storing elements with reverse deltas, the base level is the current level.

base library

The *base library* is a partitioned data set (PDS), a CA Panvalet, a CA Librarian, or ELIB file, or a USS directory that stores the base members for elements defined to CA Endeavor SCM. A base library is defined for each element type, but it can be shared across types.

base regression

The *base regression* is the percent by which the statements stored in a new level of an element change the statements stored in the base level.

batch

A *batch* is an IBM term referring to an environment in which noninteractive programs are executed. In CA Endeavor SCM, batch refers to the execution of actions and reports in a noninteractive region, as opposed to execution in the foreground.

browse

The ISPF *browse* mode lets you view the content of a data set, but does not let you change the content. In CA Endeavor SCM, the foreground Display function lets you browse CA Endeavor SCM elements. The Print element option browse lets you print the element source in batch.

Bulk Data Transfer (BDT)

The *Bulk Data Transfer (BDT)* is an IBM transmission utility supported by the CA Endeavor SCM package shipment utility. BDT is abbreviated as BDT2 for BDT version 2 and BDTN for BDT with NJE/NJI.

C1DEFLTS table

The *C1DEFLTS Table* is the CA Endeavor SCM Defaults Table.

CA Endeavor Parallel Development option

The *CA Endeavor Parallel Development Option* lets users compare and integrate three versions of source code, resolving conflicts resulting from concurrent development or from applying vendor updates to applications that have been customized in-house.

CA Endeavor SCM listing libraries

CA Endeavor SCM listing libraries are libraries used to store compressed compiler listings produced by processors.

CA Endeavor SCM location

The *CA Endeavor SCM location* refers to the stage and environment where an element resides.

Cast

The *Cast* command freezes the actions included in the package. After a package is cast, it cannot be edited and only approvers can work with it.

CCID definition data set

The *CCID definition data set* identifies the CCIDs to be used within CA Endeavor SCM. The definition file must be a card-image data set (80-byte, fixed-format records).

CCID validation

The *CCID validation* checks a CCID specified on an action against the CCIDs defined in the CCID definition data set.

Change Control ID (CCID)

A *Change Control ID (CCID)* is a logical grouping mechanism by which user-specified portions of the CA Endeavor SCM inventory can be tagged, and then viewed, tracked, and manipulated. CCIDs are optional but may be required on a system-by-system basis. The same is true of comments.

change regression

The *change regression* is the percent by which the statements stored in a new level change the revisions made by the previous level.

checksum

The *checksum* is an internally calculated value within a package. CA Endeavor SCM uses the checksum to determine if a package has been changed.

classification

Classification is the System, Subsystem, and Type associated with an element and is used in conjunction with the element's location (Environment and Stage).

comment

A *comment* is a user-defined remark (up to 40 characters) associated with an action or package. A comment is used with CCIDs, and generally describes the reason for the action or purpose of the package.

Commit

The *Commit* command records all events related to a package, and removes all backin and backout information. After a package is committed, the package can no longer be backed out or backed in.

Committed

Committed is the status of a package after the successful completion of the Commit command.

complementary data sets

Complementary data sets can be shipped along with package shipments. The complementary data sets for a shipment contain a backout of the shipment.

component

A *component* (primarily for the CA Endeavor Automated Configuration option) is the output produced or the input read in by a generate or move processor. The components of a generated element include the input components that were included to produce an output of the generate processor, the element itself, all outputs created by the generate or move processor, and user-defined data. Components are referenced by element or member names.

component list

A *component list* is a list of all components created or read by a generate or move processor. The component list can be viewed using the Print Elements action, or the Display Element/Component panel (CA Endeavor Automated Configuration option). The component list provides an audit trail of program-component information at the time of each compilation.

component monitoring

Component monitoring is a feature of the CA Endeavor Automated Configuration option that allows you to check selected data sets for component relationships.

component validation

Component validation occurs when a package is cast. CA Endeavor SCM then validates that all dependent components are in the package and that those components have not changed since they were last used.

Concurrent Action Processing

The *Concurrent Action Processing* facility causes certain element action requests to be executed concurrently, which reduces the elapsed time it takes to process multiple actions and, therefore, speeds processing of batch jobs and packages. This option must be enabled by the CA Endeavor SCM administrator.

Concurrent Number

The *Concurrent Number* parameter, on the Submit Package panel, lets you specify the number of started tasks to be initiated, when you submit a package for processing using the Concurrent Action Processing facility. This parameter overrides the default set by the SPAWNCNT value in C1DEFLTS.

configuration management

Configuration management is the capture and storage of program-component relationships and the tracking of these relationships over time.

CONNECT:Direct

CONNECT:Direct is a network transmission utility provided by Sterling Commerce and supported by the CA Endeavor SCM package shipment utility. It is abbreviated as NWD when establishing destinations. CONNECT:Direct was formerly known as Network DataMover.

consolidation

A *consolidation* is a CA Endeavor SCM facility that allows you to specify a number of delta levels to retain when a member reaches the consolidation level specified for its type.

Copy action

The *Copy action* copies an element from an archive data set to a data set external to CA Endeavor SCM.

copyback

When *copyback* (also known as fetch) is specified, CA Endeavor SCM searches for an element along the map (beginning at a designated stage), finds the element, and then copies it back to the initial stage. Copyback is available as an explicit option with the Generate Elements action. Copyback is also used when adding, transferring, and moving elements.

create a package

The *Create/Modify* function in foreground lets you create a package, specifying package-related information such as a package ID, an execution window, and so on.

current level

The *current level* is the most recent source for an element. When using forward deltas, the current level of an element comprises the base level plus all subsequent change levels. When using reverse deltas, the current level of an element is the current source.

data set validation

Data set validation is an optional feature to verify that retrieved elements are added or updated from the data set to which they were last retrieved. This ensures that the same copy of the element (revised as appropriate) is placed back in CA Endeavor SCM. Data set validation can be specified separately by system. If this facility is in effect for a system, you can override it to add the element back from a different data set if you have proper authority.

Defaults table

The *Defaults Table* (also known as C1DEFLT) is a table of site-specific information necessary for CA Endeavor SCM operation. The Defaults Table includes environment and stage definitions, installed options, and site-specific hardware settings. There is only one Defaults Table for each site.

Delete action

The *Delete action* erases base and delta forms of an element and removes element information from a Master Control File (MCF) or a component list.

delete processor

A *delete processor* is a processor that is run when an element is deleted from a stage. Typically, the delete processor deletes the output created by the corresponding generate processor.

delta level

A *delta level* is a record of a change to the base level of an element. Each change to an element creates a delta level. CA Endeavor SCM compares the current level to the new source and builds a delta level containing the changes to the source.

delta library

A *delta library* is a partitioned data set (PDS), a CA Panvalet, a CA Librarian, or an ELIB file that stores the delta members for elements defined to CA Endeavor SCM.

Denied

A *Denied* status of a package is when the package has been reviewed, but denied, by an approver.

Deny

The *Deny* command is an option for a package approver. If one approver denies a package, it cannot be executed.

destination

A *destination* is where package outputs are shipped. A destination record contains the information needed by CA Endeavor SCM to ship package outputs to the destination.

Display action

The *Display action* is used to view environment definitions, element information, and footprint-related data.

DSN mapping rule

A *DSN mapping rule* is a user-defined correspondence between host and remote data set names. DSN mapping rules are used when shipping package outputs.

element

An *element* is a partitioned data set (PDS) member, a CA Panvalet, CA Librarian, sequential data set, or USS file that has been placed under control of CA Endeavor SCM.

element change

Element changes can be reviewed in the view Display changes current level. This view of element information shows the current level of an element, annotated to indicate the level at which each line was added to the source. This view is accessible in batch using the Print element,option changes SCL.

element component

The *element component* is the part of a component list referred to as element information. This information includes the footprint of the CA Endeavor SCM source.

element history

Element history can be reviewed in the view Display history current level. This view of element information shows all lines that have been present in a piece of source code, annotated to show the level at which the line was added, deleted, or both from source.

element master

Display *element master* info is a view of Master Control File (MCF) information about an element. This view is accessible in batch using the Print Element option.

element name

The *element name* is the name assigned to an element, used to identify the element within CA Endeavor SCM.

element summary of levels

Display summary of levels is a summary view of activity against an element at all levels. Information provided includes the number of statements at each level, the number of lines added, and the number of lines deleted. This view is accessible in batch using the Print Element option summary SCL.

ELIB

An *ELIB* (Endevor LIB) is a high-performance alternative to OS-partitioned data sets under CA Endevor SCM. ELIB data sets can reorganize member space automatically as members are rewritten or deleted, exploit 31-bit storage for VSAM-organized data sets, expand directories and data sets automatically, provide improved directory processing, and maintain additional statistical information about member size.

emergency approval

An *emergency approval* is used to approve an emergency package. An approver group must be assigned the authority to approve emergency packages.

emergency package

An *emergency package* is one of two types of packages (either standard or emergency when created). Emergency packages require approval from emergency approver groups.

enable backout

The *enable backout* option appears when creating a package. You can decide whether or not to allow the package to be backed out.

Environment

An *environment* is the top level of the logical structure used to classify elements in CA Endevor SCM. Environments usually correspond to functional levels in an organization (for example, development, quality assurance, and production). Each environment has two stages. There is no limit to the number of environments you can use.

Environment name

The *environment name* is a name to identify each environment within CA Endevor SCM. The name can be up to eight characters, and include any of the following: A-Z, 0-9, @, #, and \$.

Environment title

The *environment title* is a descriptive title assigned to each environment that is used in various displays and reports (up to 40 characters).

ESI (External Security Interface)

The *External Security Interface (ESI)* is an optional interface used to implement external security at your site. When installed, this interface replaces the native security facility supplied with installation files (and implemented through the security tables), with calls to RACF, CA ACF2, or CA-Top Secret.

execute a package

The *execute* command runs a package. Packages that execute successfully can be backed out or committed.

Execution Report

The *Execution Report* is output when you run CA Endeavor SCM actions. The Execution Report documents the actions requested and the processing that occurred. The report can be viewed online by browsing data sets userid.C1TEMPR1.MSGS or userid.C1TEMPR2.MSGS. CA Endeavor SCM prints the report on DDNAME C1MSG1 on the batch SYSOUT.

execution window

The *execution window* is a timeframe (start and end date) within which a package must be executed.

exit

An *exit* is part of the CA Endeavor SCM exit interface designed for use with exits written in either assembler or in high-level languages such as COBOL.

export a package

The *export* command copies package SCL into an external data set.

External Security Interface (ESI)

The *External Security Interface (ESI)* is an optional interface used to implement external security at your site. When installed, this interface replaces the native security facility supplied with the installation files (and implemented through the security tables), with calls to RACF, CA ACF2, or CA Top Secret.

footprint

A *footprint* is encrypted data added by processors to individual source, object, or load modules, to identify the CA Endeavor SCM element associated with the module. CA Endeavor SCM uses the data to display and process information for the element. A footprint includes the site ID, environment name, stage number, system name, subsystem name, element name, element type, element version or level, and the date and time the footprint was assigned, all in encrypted format.

foreground

The *foreground* is an IBM term referring to an environment in which interactive programs are executed. In CA Endeavor SCM, you run actions in the foreground by requesting those actions through the CA Endeavor SCM Foreground Options Menu.

forward delta

Forward delta is a method for recording changes that stores a base version of code, and then builds current versions by applying changes made to the base.

forward recovery

Forward recovery is the process of taking an old level of an element and making it the current one (new level). As a result, any changes made by the levels between are backed out. To perform forward recovery, you first retrieve the older (to-be-recovered) level, and then add or update the element using the retrieved source to create the new level.

Generate Action

The *Generate action* translates source into executables, and then populates output libraries with the executables by executing the generate processor for an element.

generate processor

A *generate processor* translates source into executables, and then populates output libraries with these executables.

group name

A *group name* is a name associated with a security configuration that applies for multiple users, found within the definition of the Access and User Security Tables. The name is then associated with any number of specific user IDs, to associate the IDs with the group-level security. This is a convenient way to assign security to several users having identical levels of access to the CA Endeavor SCM environment.

identify record (IDR)

An *identify record (IDR)* is an IBM term for a record in a load module that contains user-defined data. An IDR is created by the linkage editor when it encounters an IDENTIFY statement in the object deck. Within CA Endeavor SCM, IDRs are used to store the footprints associated with load modules.

IDR (identify record)

An *identify record (IDR)* is an IBM term for a record in a load module that contains user-defined data. An IDR is created by the linkage editor when it encounters an IDENTIFY statement in the object deck. Within CA Endeavor SCM, IDRs are used to store the footprints associated with load modules.

import a package

The *import* command is used to create a package by copying SCL from an external data set.

Include library

An *Include library* is a CA Panvalet library, CA Librarian library, or partitioned data set (PDS) that contains INCLUDE members referenced within CA Endeavor SCM elements. This library is optional and it can be defined for each element type. The Include library is used by Retrieve actions if you specify that you want to expand Include members at the time the element is retrieved. It is also used by the CONWRITE utility, if you specify that you want to expand Include members during CONWRITE processing.

input component

An *input component* is an included component to produce an output when executing the generate or move processor when using the CA Endeavor Automated Configuration option. For example, a copybook is the input component when compiling a COBOL program.

inventory

An *inventory* contains the software components that make up your application software systems.

Inventory Analyzer

The *Inventory Analyzer* is a CA product that allows you to analyze your software inventory, classifying it according to CA Endeavor SCM types. This product can be used when implementing CA Endeavor SCM.

inventory area

The *inventory area* is a subset of a software inventory, defined by its CA Endeavor SCM location (environment and stage) and classification (system, subsystem, and type).

jump

A *jump* occurs when moving an element from stage 2 in one environment to a stage in another environment on a map route, when a version of the element exists at an intermediate stage that is not part of the map route.

last action

The *last action* is the most recent action executed for an element. Once executed, each action is recorded as the last action except Archive, Delete, Display, List, and Print.

last action CCID

The *last action CCID* is the CCID specified for the last action executed against an element.

level

A *level* is the source for an element at a particular time. When an element is first added to a stage, there is one level (known as the base level). Each time CA Endeavor SCM actions change the source thereafter, a new delta level is created.

level number

The *level number* is an identifier for a specific level of an element. CA Endeavor SCM assigns each set of changes a level number that is one higher than the number assigned to the preceding level.

library management

Library management is the classification, control, and storage of the physical components of a software inventory.

List action

The *List action* lists elements from a Master Control File (MCF) or archive data set in the form of action requests or members from a library. This action can also be used for text scanning. If the Automated Configuration Manager facility is installed, this action can also search a component list based on specified criteria.

load utility

The *load utility* is a CA Endeavor SCM utility that is used to load members from an external data set into any stage in an environment.

map

A *map* is the promotion routes established for software inventory at a site. Environments and stages are mapped to each other in the CA Endeavor SCM Defaults Table. Systems, subsystems, types, and processor groups are mapped to each other on their respective definition panels.

Master Control File (MCF)

The *Master Control File (MCF)* is a CA Endeavor SCM file that contains the definitions of stages, systems, subsystems, element types, and elements. This file is accessed and updated by CA Endeavor SCM to manage the element definitions, to execute processors, and for other miscellaneous functions. There are two Master Control Files (MCFs) for each environment—one per stage.

MAXRC

The *MAXRC* is a processor keyword that defines the highest acceptable return code for a processor step. If a step exceeds this return code, the CA Endeavor SCM return code (NDVR RC) is set to 12. When this happens, the Element Master display shows *FAILED* in the NDVR RC field.

MCF (Master Control File)

The *Master Control File (MCF)* is a CA Endeavor SCM file that contains the definitions of stages, systems, subsystems, element types, and elements. This file is accessed and updated by CA Endeavor SCM to manage the element definitions, to execute processors, and for other miscellaneous functions. There are two Master Control Files (MCFs) for each environment—one per stage.

model transmission control statements

Model transmission control statements control the functioning of data transmission programs used by the package shipment utility.

Move action

The *Move action* moves elements between stages, within or across environments.

move processor

Move processors copy outputs, element information, and component lists from the source location to the target location of a Move or, optionally, a Transfer action.

name masking

Name masking lets you substitute a wildcard asterisk (*) character or a placeholder percent sign (%) character for one or more characters in a search string. Name masking is valid on many, but not all search requests.

Native Security

Native Security provides functional security by controlling environment access, primary and foreground menu options, and action authorization. When security control points are reached, CA Endeavor SCM checks access privileges defined in the Native Security tables. This is one of two methods of providing functional security; the administrator must choose to implement functional security using either the Native Security feature or the CA Endeavor SCM External Security Interface (ESI) option.

NDVR RC (return code)

The *return code (NDVR RC)* results from action processing. Values can be 00, 04, 08, 12, and 16.

NoSource action option

NoSource is an option for the Generate action that eliminates the need to fetch an element's source back to the target location when the element does not already exist at the target. This option can be specified on foreground or batch Generate actions.

notification facility

The *notification facility* is a CA Endeavor SCM facility that allows you to notify users of events that require a response from them.

output component

An *output component* is created as a result of executing the generate or move processor when using the CA Endeavor Automated Configuration option. For example, an object deck is an output component when compiling a COBOL program.

output library

An *output library* is any of several libraries used during output management, including the CA Endeavor SCM processor listing library, processor load library, and source output library as well as user copy libraries, load libraries, listing libraries, macro libraries, JCL libraries, databases, and so forth.

output management

Output management is an aspect of CA Endeavor SCM that handles the creation and maintenance of various outputs that relate to an element. The exact nature of these outputs varies depending on the corresponding element type, and is defined by the output management for that type.

package

A *package* is a group of CA Endeavor SCM actions that requires approval before it can be executed. Creating packages allows you to group specific actions so they can be maintained and tracked as a single unit, establish formal approval procedures to ensure data integrity through modifications, and centralize specific action groups so you can see them across environments and reuse them.

package data set

A *package data set* is where packages are stored. There is one package data set for a site.

package events

The *package events* are an audit trail recording the events that have occurred involving a package, logged by user ID, date, and time. Package events relate to the various steps of the package processing procedure, and include Created, Last Updated, Cast, Approved, Executed, Backed Out, Backed In, and Committed.

package exits

A *package exit* is called before, after (or both) package functions and subfunctions.

package shipment

A *package shipment* is the transmission of package outputs, and optionally their backouts, from host sites to remote sites.

package status

A *package status* indicates the status of a package at any specific time. Status levels for packages include In-edit, In-approval, Denied, Approved, In-execution, Executed, Exec-failed, and Committed.

Print action

The *Print action* is a batch command that prints element or member information.

PROC RC (processor return code)

The *processor return code (PROC RC)* is the highest return code from the execution of a processor. This return code is set to *FAILED* if the return code for any step in a processor exceeds the MAXRC for the processor. This return code is set to *PROC'D?* if the element has not been generated after being restored or transferred from an archive data set or added or updated.

PROC statement

A *PROC statement* is a job control statement used in catalogued or in-stream procedures. PROC statements can be used to assign default values to symbolic parameters contained in a procedure. A PROC statement is also used to mark the beginning of in-stream procedures.

processor component

A *processor component* is the part of a component list that includes processor information when using the CA Endeavor Automated Configuration option. This information includes the footprint of the processor.

processor group

A *processor group* identifies a set of processors for a specific element type, as well as the default symbolic overrides for the processors' JCL. A group can include up to three processors (one generate, one delete, and one move processor, or any combination). Processor groups are useful when elements of one type require slightly different processing. When you define a type to CA Endeavor SCM, you can also identify a default processor group for that type. Using symbolics when writing the processors for the default processor group can allow you to use the same processors, by changing symbolic definitions, for other processor groups associated with this type.

processor group symbolics

Processor group symbolics are defined in PROC statements in one or more processors in a processor group. The symbolics and their default values appear on the Processor Group Symbolics panel. By modifying the default values, you can use one processor in more than one processor group.

processor listing library

The *processor listing library* is an optional library that stores the listings output from the CA-supplied processor named GPPROCSS.

processor load library

The *processor load library* is a CA Endeavor SCM library that contains the load-module form of each processor. The modules from this library are executed when processors are invoked.

processor output library

The *processor output library* is a library (referred to in a processor) to which the processor writes output. Processor output libraries can be source libraries, executable libraries, or listing libraries.

processor return code (PROC RC)

The *processor return code (PROC RC)* is the highest return code from the execution of a processor. This return code is set to *FAILED* if the return code for any step in a processor exceeds the MAXRC for the processor. This return code is set to *PROC'D?* if the element has not been generated after being restored or transferred from an archive data set or added or updated.

processors

A *processor* is a standard OS JCL job stream that manipulates elements and their outputs. There are generate, delete, and move processors. CA Endeavor SCM supports both CA Endeavor SCM symbolics and user-defined symbolics in processors. This capability allows you to write one processor that you can use in multiple processor groups by changing the values assigned to one or more symbolics. CA Endeavor SCM also provides a set of utilities for use when writing processors, and supports the use of in-stream data in processors.

production data set

The *production data set* is used to store production code. This term is used in the package shipment utility to refer to host and remote production data sets.

program pathing

Program pathing is a security option under RACF, CA ACF2, and CA Top Secret, that allows you to restrict the data sets available to particular users, as well as the programs and load libraries from which those data sets can be accessed. This is not a CA Endeavor SCM option, but is specific to RACF, CA ACF2, and CA Top Secret.

promote

Promote is another term for moving an element from one inventory area to another inventory area.

promotion management

Promotion management is the task of coordinating and validating successive changes to the various inventory areas in a software development setting.

promotion package

A *promotion package* is a type of package that contains move actions only and the from environment and stage location for all move actions must be the same. A promotion package can be reused until the contents have been promoted to the end of the map, without having to reset the package, manually edit the SCL to modify the inventory source location, or re-cast the package. However, a promotion package needs to be approved between executions, if required.

quorum

A *quorum* is the minimum number of approvers for which approval is required to execute a package. When a quorum size is specified, at least that many approvers must review and approve a package.

regression

Regression refers to the condition where one set of changes to element source is overwritten by a subsequent set of changes. CA Endeavor SCM flags regression when the changes stored for a specific level of an element overwrite more than a predefined percentage of the element statements.

regression percent

The *regression percent* is the percent of acceptable change to element source, defined for each element type. When this percent is exceeded, the result is a user-specifiable CA Endeavor SCM return code (0, 4, 8, or 12). Each time a new level is created for an element, CA Endeavor SCM checks the changes stored in that level against this percent, changes to both the statements stored for the base level (known as base regression) and changes to the statements stored for the previous level (known as change regression). If the amount of change in either case exceeds the defined percent, CA Endeavor SCM issues a message of user-defined severity.

Reload utility

The *Reload Utility* allows you to recover a CA Endeavor SCM VSAM control file (Master Control File, package data set) or a base or delta data set that was lost because of a physical device failure or site disaster. The RELOAD action restores data from data sets created by the unload process.

remote footprint synchronization

Remote footprint synchronization is a procedure in which footprinted executables are shipped from a remote site to a host site, where footprint reports are run to compare the executables' footprints with host Master Control File (MCF) information.

remote nodename

A *remote nodename* is part of a package shipment destination that identifies the site to which package outputs are to be shipped. The name must be valid for the selected data transmission program.

request data set

A *request data set* contains action requests to be submitted for batch processing. You create request data sets in foreground using the SCL Generation facility.

request for data

The *request for data* is a package exit capability allowing package exit programs to make multiple, successive requests for CA Endeavor SCM information on a single invocation of the exit.

reset a package

The *reset a package* action erases all package event records, returning the package to In-edit status.

Resource Security Table

The *Resource Security Table* defines the element names that are restricted to a particular system and subsystem, within a specific environment. The Resource Security Table is defined by the CA Endeavor SCM administrator using the CONSDEF macro. Each CA Endeavor SCM environment can contain, at most, one Resource Security Table.

Restore action

The *Restore action* restores an element to CA Endeavor SCM from an archive data set.

Retrieve action

The *Retrieve action* copies any level of an element to an external data set.

Return Code (NDVR RC)

The *return code (NDVR RC)* results from action processing. Values can be 00, 04, 08, 12, and 16.

reverse delta

The *reverse delta* is a method for recording changes that stores the most recent version of the code, rebuilding prior versions by backing out individual changes from the current version.

review a package

A *review a package* is performed to review the contents of the package. After reviewing a package, an approver approves it or denies it.

route

A *route* is a series of environment and stage locations that make up the stages in a software life cycle. Working together, all the routes at a site constitute the map for that site.

SCL (Software Control Language)

Software Control Language (SCL) is the CA Endeavor SCM language that is used in batch processing to maintain or act against elements within CA Endeavor SCM.

security

Security is a CA Endeavor SCM feature that allows you to restrict action requests and access to elements. The security system supplied with CA Endeavor SCM (known as native security) is implemented using three tables.

sharable

A *sharable* package can be edited by users who did not create the package.

ship utility

The *ship utility* is a CA Endeavor SCM utility that allows you to ship package outputs to remote sites.

shipment confirmation

Shipment confirmation occurs in a package shipment after executing the data transmission utility and after executing the remote copy or delete job step.

shipment staging

Shipment staging is the creation and population of host staging data sets with package outputs or backout members.

Signin action

The *Signin action* removes the current signout for an element. Signin can be implicit or explicit.

signout

A *signout* is the assignment of a user ID to an element, establishing ownership of the element. Signout is automatic when adding or updating elements in, or when retrieving elements from, CA Endeavor SCM.

site

The *site* is the location at which CA Endeavor SCM is installed. The site is defined in the CA Endeavor SCM Defaults Table, where it is assigned a site ID.

Software Control Language (SCL)

Software Control Language (SCL) is the CA Endeavor SCM language that is used in batch processing to maintain or act against elements within CA Endeavor SCM.

software lifecycle

The *software lifecycle* defines the stages through which software passes at a site during development and maintenance. A software lifecycle may consist of development, testing, quality assurance, and production.

software management

Software management is the process of tracking changes to software components and their interrelationships over time. This process includes configuration management, library management, software distribution, and version control.

source

The *source* is the non-executable form of an element.

source management

Source management is the aspect of CA Endeavor SCM that deals with the creation and maintenance of element source. Element source is maintained in base and delta libraries in an internal format or in standard IBM format (if reverse deltas and non-encryption are selected).

source output library

A *source output library* is a CA Endeavor SCM library that contains the latest full source version of each element. This library is designed for use with copybooks, macros, procedures, and so forth that are copied to other places. This is an optional library. However, if used, it is specified in the definition of the corresponding element type.

sourceless element

A *sourceless element* is created when the Generate NoSource action does not find the element source at the target location and then uses the first element source found up the map from the target to generate the element at the target. Because the element source is not fetched back to the target location, the MCF record for the element generated at the target identifies it as a sourceless element. The MCF element contains the last level timestamp of the upstream element. Base and deltas do not exist for the sourceless elements, so these fields are blank in the MCF. When actions are performed against sourceless elements, the source from the next sourced element upstream from it is used, providing that the last level timestamps are equal.

stage

A *stage* is a step in the software lifecycle. There are two stages defined for each CA Endeavor SCM environment.

stage ID

The *stage ID* is an identifier for the stage that is used when processing an element action to select (identify) the stage you want to process.

stage name

The *stage name* is assigned to each stage during CA Endeavor SCM installation. A stage name can include any of the following characters: A-Z, 0-9, @, #, and \$.

stage number

The *stage number* (either 1 or 2) is the relative number for the stage within the environment.

stage title

The *stage title* (up to 20 characters) is assigned to each stage and is used in displays and reports to describe the stage.

standard approval

A *standard approval* is one of two types of package approval. Standard packages can be approved only by standard approver groups.

standard package

A *standard package* is one of two types of packages (standard or emergency). Standard packages require approval from standard approver groups.

STOPRC

The *STOPRC* value provides a control for processing during batch execution by setting your highest acceptable return code for the action that is currently processing. Batch processing will terminate if the return code for the executed action exceeds the *STOPRC* value. The *STOPRC* value is set by the statement SET *STOPRC* nn specified as input to the SCL run. Also, the *STOPRC* value can be set from the Batch Options Menu when building SCL in foreground for execution in batch, and it can be set on the Create/Modify Package panel when creating or modifying a package. Valid values are 4 to 16. The default is 16.

subsystem

A *subsystem* is a part of the CA Endeavor SCM classification of an element. Subsystems are used for specific applications within a system. For example, you may have a purchase order and accounts payable subsystem within the financial system.

symbolic

A *symbolic* is used by CA Endeavor SCM processors to represent a value specific to an individual run of a processor. CA Endeavor SCM supports two kinds of symbolics (CA Endeavor SCM symbolics and user symbolics). CA Endeavor SCM symbolics (preceded by &C1) are used in CA Endeavor SCM processors to represent a value specific to an individual run of the processor. Values are assigned to CA Endeavor SCM symbolics when the processor is executed. User symbolics are defined by users in JCL PROC statements in processors. You can use one processor in multiple processor groups.

synchronize

A *synchronize* conflict is detected when transferring or moving an element with history, if the current level of the target does not match any level of the source. CA Endeavor SCM searches for the level of the target to match a level of the source, and this level becomes the synchronization level. When there is a synchronization conflict, CA Endeavor SCM does not allow the element to be transferred or moved unless the synchronization flag (SYNC option) is set to Y.

system

A *system* is a way to classify elements in CA Endeavor SCM (defined to each environment where it will be used). A system typically represents the applications at a site. For example, you may have financial and manufacturing applications.

System Management Facilities (SMF) interface

The *System Management Facilities (SMF) interface* is an optional interface to IBM's System Management Facilities (SMF) that allows you to record historical information through SMF records. In CA Endeavor SCM, they are called Action Records or Security Records. This information is used to generate historical reports. The implementation of the SMF interface is optional at each site, and the recording of historical information is optional within each environment.

Transfer action

The *Transfer action* transports elements from a source location to a target location. Each location can be a CA Endeavor SCM location or an archive data set.

transmission method

The *transmission method* is a part of package shipment destinations to identify the transmission utility to be used to ship packages to the destination.

transportable footprint

A *transportable footprint* is a CA Endeavor SCM footprint that can be imbedded in DOS/VSE- and VM/CMS-bound object modules using a z/OS compiler, a VSE compiler, or a VM compiler.

type

A *type* is a category of source code that is used as part of the classification of a CA Endeavor SCM element. For example, there are types for COBOL (COBOL code), COPYBOOK (copybooks), JCL (JCL streams), and so forth.

type processing sequence

A *type processing sequence* is a relative sequence of processing for the element types defined to each system. By defining a processing sequence, for example, you can make sure that copybooks (type COPYBOOK) are updated before any COBOL programs (type COB) that may use the copybooks.

Unload utility

The *Unload Utility* unloads and validates the contents of the VSAM Master Control Files (base and delta files associated with the environments and systems specified on the job request). The file created by the Unload function contains a backup of all internal MCF definitions (system, subsystem, type, type sequence, data set, element master record) and base or delta data (element base, element delta, component base, component delta). Packages contained within a package data set can also be unloaded. The utility can be run for an entire environment or for selected systems within an environment. Unload may also be directed to back up an entire package data set or individual packages.

Update action

The *Update action* adds a member to CA Endeavor SCM when an element with the same name is located in the target entry stage.

User Exit table

The *User Exit Table* identifies exit programs to be called at each CA Endeavor SCM exit point.

user ID

The *user ID* is used in TSO for actions that run in foreground for the session. For those actions run in batch, the user ID is the job name or the ID specified through the USER= parameter on the job card, depending on how your CA Endeavor SCM Defaults Table is set up.

user menu facility

The *user menu facility* allows your CA Endeavor SCM administrator to attach user-defined functions to the CA Endeavor SCM TSO/ISPF front end.

User Security table

The *User Security Table* defines the systems and subsystems to which users have access, and for each system and subsystem, the type of processing (authorization level) allowed. Each environment has one User Security Table.

using elements

A *using element* is an element that uses a component element. For example, programs that use copybook COPYA are known as the using elements for COPYA. For more information, see the [Autogen action option](#) (see page 116).

validate

The *validate* function allows you to ensure the integrity of one or more existing CA Endeavor SCM environments and systems, and their related elements and components. These are the same checks performed as part of Unload processing, allowing this function to operate in a stand-alone mode.

Validate action

The *Validate action* performs element master, synchronization, and component validation checks against the element you specify. These inventory validation checks can be performed at any time, enabling developers to ensure the integrity of elements before including them in a package. This is a batch only action and is not supported in a package.

version

The *version* is a two-digit identifier associated with an element. You cannot have two versions of an element in the same environment.

version control

Version control is the maintenance, tracking, and auditing of modifications to an application over time, allowing prior development versions to be restored.

version number

The *version number* identifies the version assigned to an element.

Index

A

actions

- adding elements • 30, 34
 - deleting elements • 47, 48
 - execution report message • 26, 27
 - generating elements • 48, 50
 - generating elements with copyback • 51
 - generating elements without copyback • 52
 - moving elements • 39, 45
 - moving elements with history • 41
 - moving elements without history • 43
 - packaging elements • 61
 - printing elements • 54, 55
 - retrieving elements • 27, 29
 - signin elements • 57, 58
 - updating elements • 35, 38
 - viewing approver group definitions • 86
 - viewing approver group relationship definitions • 87
 - viewing element information • 77
 - viewing environment information • 88
 - viewing footprint information • 81
 - viewing global type processing sequence • 89
 - viewing processor group definitions • 85
 - viewing site definitions • 82
 - viewing site symbolics • 88
 - viewing stage information • 82
 - viewing subsystem definitions • 83
 - viewing system definitions • 83
 - viewing type definitions • 84
- ### adding and updating elements
- processor groups • 97
- ### adding elements • 30, 34
- CCIDs and comments • 34
 - processor groups • 96
 - signout status • 33
- ### application developers • 9
- ### approver group definitions
- viewing • 86
- ### approver group relationship definitions
- viewing • 87
- ### audience
- application developers • 9
 - development managers • 9

- quality control engineers • 9
- quality control managers • 9
- source control managers • 9

Autogen • 48

C

CA Endeavor SCM

- exiting • 13
- starting • 13

CCIDs and comments

- adding elements • 34
- generating elements • 54
- moving elements • 44
- retrieving elements • 29
- signin elements • 57
- updating elements • 38

commands

- EONLY • 16
- EPRINT • 17
- ESORT • 16
- ISPF • 21
- stacking • 22

copyback

- generating elements • 51
- generating elements without • 52

D

dates

- selection lists • 18

default values

- fields • 14

defaults, user preferences • 23

deleting elements • 47, 48

- processor groups • 98

development managers • 9

development tasks

- daily • 25
- periodic • 25

documentation

- displaying • 14

E

elements

- adding • 30, 34

- deleting • 47, 48
- execution report message • 26, 27
- generating • 48, 50
- generating with copyback • 51
- generating without copyback • 52
- moving • 39, 45
- moving with history • 41
- moving without history • 43
- packaging • 61
- printing • 54, 55
- retrieving • 27, 29
- signin • 57, 58
- updating • 35, 38
- viewing information • 77
- emergency operations
 - software life cycle • 11
- environment information
 - viewing • 88
- EONLY command • 16
- EPRINT command • 17
- ESORT command • 16
- execution report message • 26
 - viewing • 27
- Execution report preferences • 23
- exiting CA Endeavor SCM • 13

F

- fields
 - default values • 14
- footprints
 - viewing information • 81
- foreground option preferences • 23

G

- generate processor
 - moving elements • 40
- generating elements • 48, 50
 - CCIDs and comments • 54
 - processor groups • 99
 - signout status • 53
- generating elements with copyback • 51
 - processor groups • 101
- generating elements without copyback • 52
- global type processing sequence
 - viewing • 89

H

- help

- displaying • 14
- history
 - moving elements • 41
 - moving elements without • 43

I

- input files
 - trailing blanks • 37
- ISPF
 - overview • 9
- ISPF commands • 21

J

- job statement preferences • 23
- JUMP and RETURN key preferences • 23

K

- keyboard
 - ISPF commands • 21

L

- life cycle
 - software • 10, 11
- lines per page preference • 23
- long element names
 - selection lists • 17

M

- mainframe
 - environment • 9
 - ISPF facility • 9
 - operating system • 9
 - TSO facility • 9
- messages
 - execution report • 26, 27
- moving elements • 39, 45
 - CCIDs and comments • 44
 - processor groups • 103, 105
 - signout status • 44
- moving elements with history • 41
- moving elements without history • 43

N

- name masking • 19
- navigation
 - ISPF commands • 21
 - stacking commands • 22

NoSource • 48, 52

O

operating system
 overview • 9
outputs
 creating • 48, 50

P

package
 casting • 62
 creating • 61
 executing • 64
 reviewing and approving • 63
packaging elements • 61
placeholders
 selection lists • 18
preferences, setting user defaults • 23
primary/secondary quantity preference • 23
printing elements • 54, 55
processor
 moving elements • 40
processor group definitions
 viewing • 85
processor groups
 adding and updating elements • 97
 adding elements • 96
 deleting elements • 98
 generating elements • 99
 generating elements with copyback • 101
 moving elements • 103, 105
 restoring elements • 102
 transferring elements • 108, 111

Q

quality control engineers • 9
quality control managers • 9

R

restoring elements
 processor groups • 102
retrieving elements • 27, 29
 CCIDs and comments • 29
 signout id • 29

S

selection lists

 dates • 18
 filtering • 16
 long element names • 17
 placeholders • 18
 printing • 17
 sorting • 16
 wildcards • 18
signin elements • 57, 58
 CCIDs and comments • 57
 signout id • 58
signout id
 retrieving elements • 29
 signin elements • 58
signout status
 adding elements • 33
 generating elements • 53
 moving elements • 44
 updating elements • 37
site definitions
 viewing • 82
site symbolics
 viewing • 88
software life cycle • 10, 11
 emergency operations • 11
 stages • 10, 11
source control managers • 9
sourceless element • 48, 52
space units preference • 23
stacking commands • 22
stages • 10, 11
 software life cycle • 10, 11
 viewing • 82
starting CA Endeavor SCM • 13
subsystem definitions
 viewing • 83
sysout class preference • 23
system definitions
 viewing • 83

T

transferring elements
 processor groups • 108, 111
TSO
 overview • 9
type definitions
 viewing • 84

U

- unit name preference • 23
- updating elements • 35, 38
 - CCIDs and comments • 38
 - signout status • 37
 - trailing blanks • 37

V

- Validate element action • 67
- viewing approver group definitions • 86
- viewing approver group relationship definitions • 87
- viewing element information • 77
- viewing environment information • 88
- viewing footprint information • 81
- viewing global type processing sequence • 89
- viewing processor group definitions • 85
- viewing site definitions • 82
- viewing site symbolics • 88
- viewing stage information • 82
- viewing subsystem definitions • 83
- viewing system definitions • 83
- viewing type definitions • 84
- volume serial preference • 23

W

- wildcards
 - selection lists • 18
- work dataset allocation preference • 23