IBM Configuration Management Version Control SC09-1783-00

Client/2 User’s Guide

Version 2 Release 2

IBM Configuration Management Version Control

SC09-1783-00

Client/2 User’s Guide

Version 2 Release 2

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page xi.

First Edition (Dec 1993)

This edition applies to the OS/2 client workstation feature for Version 2, Release 2, Modification Level 0, of IBM Configuration Management Version Control/6000 (Program 5765-207), IBM Configuration Management Version Control for Sun systems (Program 5622-063), IBM Configuration Management Version Control for HP systems (Program 5765-202), and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to:

IBM Canada Ltd. Laboratory
Information Development
2G/345/1150/TOR
1150 Eglinton Avenue East
North York, Ontario, Canada. M3C 1H7

You can also send your comments by facsimile (attention: RCF Coordinator), or you can send your comments electronically to IBM. See "Communicating Your Comments to IBM" for a description of the methods. This page immediately precedes the Readers' Comment Form at the back of this publication.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent product, program or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

This publication contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks and Service Marks

The following terms, denoted by an asterisk (*), used in this publication, are trademarks or service marks of International Business Machines Corporation in the United States or other countries:

AIX
Common User Access
CUA
IBM
IBMLink
PROFS
Operating System/2
OS/2
Personal System/2
PS/2
RISC System/6000

The following terms, denoted by a double asterisk (**), used in this publication, are trademarks of other companies as follows:

HP	Hewlett-Packard Company
HP-UX	Hewlett-Packard Company
NetLS	Apollo Computer Company
Network File System	Sun Microsystems, Inc.
Network License System	Apollo Computer Company
NFS	Sun Microsystems, Inc.
OSF/Motif	Open Software Foundation
PVCS Version Manager	INTERSOLV. Inc.
Sun	Sun Microsystems, Inc.
SunOS	Sun Microsystems, Inc.

About This Book

This book is part of the library that supports the IBM* Configuration Management Version Control (CMVC) licensed programs for CMVC *clients* and *servers*. It provides step-by-step information to help you work with the IBM Operating System/2* (OS/2*) client for CMVC in your daily development *environment*.

Who Should Read This Book

Application developers who want to learn how to perform specific CMVC *actions* with the CMVC *graphical user interface* in the OS/2 environment, should read this book. For instructions on how to use the GUI in the OSF/Motif** environment, refer to the book *IBM CMVC User's Guide* SC09-1634. Others who may want to read this book include project leaders, development managers, testers, product assurers, and technical writers.

What You Should Know

Read the book *IBM CMVC Concepts*, SC09-1633, before using the CMVC GUI This book introduces the fundamentals of the *configuration management*, *version control*, *change control*, and *integrated problem tracking* features of CMVC. It also defines the CMVC *actions* and establishes their interrelationships.

The CMVC licensed programs can be used with different workstations, including RISC System/6000*, HP**, and Sun** workstations. To use the CMVC programs effectively, you should be familiar with the operating system for your workstation. To use the OS/2 client, you should be familiar with the OS/2 environment.

Organization of This Book

When you are familiar with the concepts of CMVC, use this book to learn how to use the GUI and how to carry out specific actions.

The information in this book is organized as follows:

- Chapter 1, "Introduction to the Graphical User Interface" describes the windows, menu bars, and dialog boxes in the GUI.
- Chapter 2, "Using the GUI Windows" discusses how to perform basic tasks with the GUI, such as, opening windows, listing information, and navigating and performing actions in windows.
- Chapter 3, "Working with Files" explains CMVC file management actions and shows how to work with *files*.
- Chapter 4, "Working with Defects" discusses how to report problems by opening a *defect* and shows how to monitor and influence the progress of defects as they are resolved.
- Chapter 5, "Working with Features" discusses how to suggest new or enhanced functions by opening *features* and shows how to monitor and influence the progress of features as they are implemented.

- Chapter 6, “Working with Sizing Records and Verification Records” discusses how to use sizing records or verification records when opening or closing defects and features.
- Chapter 7, “Working with Tracks” discusses how to use *tracks* to monitor the progress of file changes made to resolve a defect or to implement a feature.
- Chapter 8, “Working with Levels and Level Members” discusses how to define a *level* of file changes within a *release* and shows how to identify a *level member* to represent the specific changes for that level.
- Chapter 9, “Working with Approval, Fix, and Test Records” discusses how to use approval, fix, and test records to track changes in a release.
- Chapter 10, “Working with User IDs and Host Lists” discusses how superusers can create or delete CMVC *user* IDs, and add or remove access on a *host*.
- Chapter 11, “Working with Components” explains how to perform actions with the CMVC - Components window. It also shows how to maintain *access lists* for an *authority* group and interest *notification lists* for a component.
- Chapter 12, “Working with Releases” explains how to group files together into a release and shows how to change the properties of a release.

Highlighting Conventions

The following highlighting conventions are used in this book:

Bold Commands, flags, files, directories, and other items predefined by CMVC are in **bold**. Valid abbreviations for commands are also in bold.

Italic Arguments or options whose names or values must be supplied by you are in *italics*. Italics are also used for emphasis, for the first occurrence in text of terms that are in the glossary, and for book titles.

`Monotype` Examples of specific data values, examples of text that you might see displayed, messages, menu items that start an action, or information that you should type are in `monotype`.

UPPERCASE Commands, parameters, and flags that are not case-sensitve are shown in UPPER CASE. You can type these commands, parameters, and flags in UPPERCASE, lowercase, or Mixed Case.

lowercase and Mixed Case
Commands, parameters, and flags that are shown in either lowercase of Mixed Case should be typed exactly as they are shown in the book.

Initiating Actions

The following terms are used throughout this book to refer to specific operations of the mouse and keyboard for the personal computer.

Click Press and release a mouse button once. You can position the pointer over a button in the GUI windows or dialog boxes and click on the left mouse button.

Double-Click Press and release a mouse button twice in rapid succession.

Select Position the pointer over a specified object (such as, a push button, cascade button, radio button, toggle button, or list item) on the display and click the left mouse button. The display changes to indicate your selection.

Drag Position the pointer over a specified object on the display, an object representing an action you want to perform, or an object on which actions are performed. Press and hold down the left mouse button while moving the mouse. Keep the button pressed and move the mouse until the pointer is positioned on another object on the display. Release the left mouse button. The display changes to indicate that you selected everything between and including the two anchor points.

Type Type text in an entry field.

Using CMVC without the GUI Program

If you prefer, you can enter commands directly on the command line instead of using the GUI.

For a description of command line syntax and the actions that you can perform with the command-line interface, refer to the book *IBM CMVC Commands Reference* SC09-1635. You can also view the online help for the CMVC commands by entering the following on an OS/2 command prompt:

```
view cmvccmd
```

An online book containing all the CMVC commands and syntax, background information, and examples is displayed.

Related Publications

You can find information about the CMVC licensed programs in the following publications.

IBM CMVC Library

- *IBM CMVC Server Administration and Installation*, SC09-1631, contains detailed information about planning, installing, customizing, operating, and maintaining the *CMVC server*.
- *IBM CMVC Concepts*, SC09-1633, provides the basis for your understanding of CMVC. It describes in detail the concepts and processes involved in using CMVC.
- *IBM CMVC User's Guide*, SC09-1634, describes all CMVC actions as implemented in the graphical user interface (GUI) for the OSF/Motif environment.
- *IBM CMVC User's Reference*, SC09-1597, contains the reference lists, tables, and *state* diagrams for CMVC. It also describes how the message-integrated CMVC GUI uses the *Broadcast Message Server (BMS)* system to fully integrate with other integrated development environment tools on the RISC System/6000, HP, and Sun workstations.

- *IBM CMVC Commands Reference*, SC09-1635, describes the CMVC commands, their syntax and use, as implemented in the command-line interface.
- *IBM CMVC Client Installation and Configuration*, SC09-1596, contains detailed information needed to install and configure the CMVC client for the RISC System/6000, HP, and Sun workstations.
- *NetLS Quick Start Guide*, SC09-1661, provides the information needed to set up the Network License System (NetLS) software to work with CMVC.
- *Managing Software Products with the Network License System*, SC09-1660, provides the information needed to manage the use of the NetLS software with CMVC.
- *Client/2 Getting Started*, SC09-1599, contains detailed information about installing and configuring the OS/2 workstation client for CMVC.

OS/2 Books

You may also want to refer to the following books to familiarize yourself with the OS/2 operating system:

- *OS/2 2.0 Getting Started*, 42G0236, explains the basics of the OS/2 operating system and how to use the online information provided with it.
- *OS/2 2.0 Using the Operating System*, 42G0238, helps you get started using your programs. It also provides additional information for some of the common tasks described in *OS/2 2.0 Getting Started*.
- *Object-Oriented Interface Design: IBM Common User Access Guidelines*, SC34-4399, describes the keys that you can use to perform actions in OS/2 applications. If you are not using a mouse, you may want to refer to this book.

Online Help Information

Online help information is available with the CMVC GUI. Use the online help when you need more information about a window or a dialog box.

For more information about how to access the online help information, refer to “Help Menu” on page 8.

Chapter 1. Introduction to the Graphical User Interface

The CMVC graphical user interface (GUI) is an interface that you can use to access, display, and perform actions on the CMVC objects in your development environment.

The CMVC GUI uses an object-action selection model; that is, you must first select an object and then perform an action on that object. You can access all lists and menus using either the mouse or the keyboard navigation model or both.

This chapter describes the main windows and window elements, and shows how to use GUI windows and menus. It also summarizes general conventions for navigating GUI windows and performing actions on CMVC objects.

Overview of the Windows

This section describes the hierarchy of main windows, and the relationship of other windows to the main windows.

Main Windows

The GUI main windows are ordered in a tree fashion. Figure 1 shows an overview of the main windows.

The CMVC - Tasks window shown in Figure 2 on page 3 is displayed when you start the GUI program. You can navigate all other windows using either the Windows pull-down menu (in the CMVC - Tasks window) or the Windows popup menu (in each GUI window).

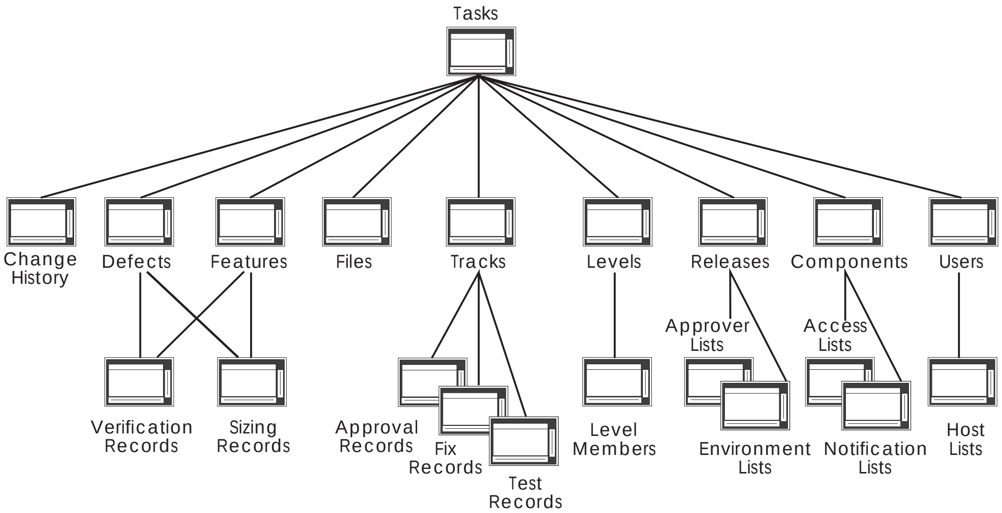


Figure 1. Hierarchy of GUI Main Windows

Other Windows

In addition to the main GUI windows, you can use other windows to display textual information and to perform certain CMVC actions. These windows include the CMVC - Change History and the CMVC - Information windows.

The CMVC - Change History window lists change history information for releases, levels, and files. To access this window, select `Change history` from the Show menu in the Defects, Features, Files, Levels, Tracks, Fix Records, or Level Members windows.

The CMVC - Information window is displayed when you request information or perform an action with the Show menu in certain GUI windows. You can use this window to review, save, or print information, and to search for related items.

For a more detailed description of how to use these windows, refer to Chapter 2, "Using the GUI Windows" on page 15.

Parts of a Window

The parts of a window and their functions are:

System Menu Contains selections for sizing, moving, or closing the window

Title Describes the main task that you can perform with the window

Menu Bar Contains a list of the various menus that you can access

Minimize Button Reduces the window to an icon

Maximize Button Enlarges the window to maximum size

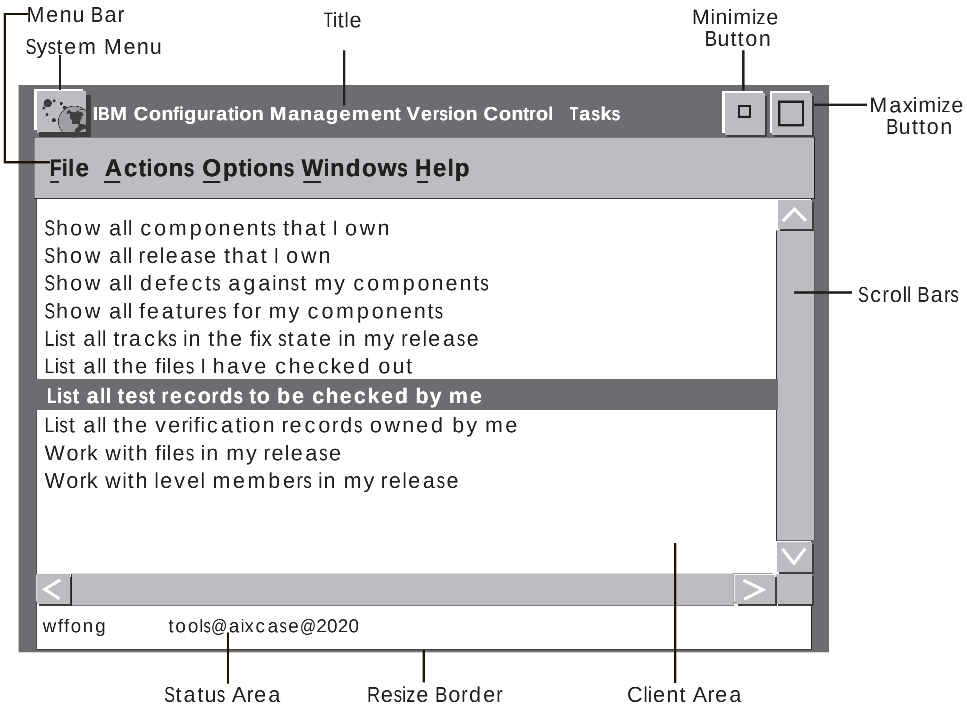
Client Area Displays lists of CMVC objects

Scroll Bar Repositions the contents of the client area so that you can view items that do not fit in the current display

Status Area Displays the family name, the host name, the user ID, the number of records displayed in the list, and the number of selected records

Resize Border Enlarges or reduces the size of the window

The parts of a window are shown in Figure 2.



Show all components that I own
Show all release that I own
Show all defects against my components
Show all features for my components
List all tracks in the fix state in my release
List all the files I have checked out
List all test records to be checked by me
List all the verification records owned by me
Work with files in my release
Work with level members in my release

wffong tools@aixcase@2020

Figure 2. Parts of a Window

Description of Menus

You can access a variety of menus from the menu bar in the GUI windows. Some menus, such as, the Files, Options, and Help menus, are in each window. Other menus such as the Actions, Modify, Windows, and Show menus are in specific GUI windows. This section describes these menus and the actions that you can perform with them.

File Menu

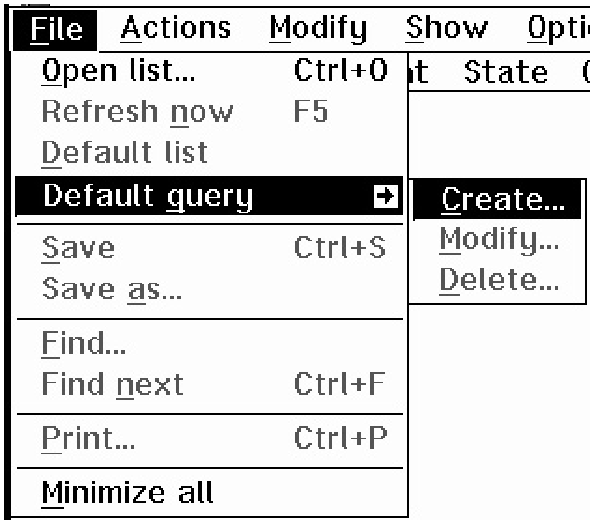


Figure 3. File Menu of a CMVC Window

This is the first menu on the menu bar in all GUI windows. Use this menu to work with queries and tasks, create reports, search for list items, and perform common file management actions. Some of the menu options and the actions that you can perform include:

Open list...
Invokes a secondary window from which you can specify a *query* to be issued to the CMVC server. The results of the query display as a list in the client area of the window. The query can be saved as a task in the Tasks List, or it can be saved as the *default query*. This option is not available in the File menu of the CMVC-Tasks window; see Figure 4 on page 5 shows the options available from the Tasks window.

Refresh now
Reissues the query for the current window and displays the query results in the client area of the window.

Default list
Issues the specified default query. The results of the query display as a list in the client area of the window. This option is not available in the File menu of the CMVC-Tasks window.

Default query
Displays a cascaded menu from which you can select `Create...`, `Modify...` or `Delete...`. Use `Create...` to define a new query for the window. Use `Modify...` to change an existing query for the window. Use `Delete` to remove the specified default query associated with the window.

Save
Saves the contents of a window to a file. If the file name is defined, the window contents are saved. If the file name is not defined, the `Save as...` dialog box is displayed. Use this dialog box to specify a file name and report format.

Save as...
Allows you to specify a file name and output format before saving the contents of a window. If you specify a file name, the name is saved and reused until you change it.

You can use the `Save as...` option to create reports in one of three formats, including the Window List, Report Table, and Report Long formats. The available formats vary depending on the window with which you are working.

Find
Finds a text string in a list of items that are displayed in the client area of the window.

Find next
Finds the next occurrence of a text string in a list of items that are displayed in the client area. The `Find next` function searches sequentially from the selected item and displays the matching item in the client area of the window.

Print...
Prints the contents of a window to the specified printer.

You can use this item to create reports in one of three formats, including the Window List, Report Table, and

Report Long formats. The available formats vary depending on the window with which you are working.

Minimize all Reduces all displayed GUI windows to icons.

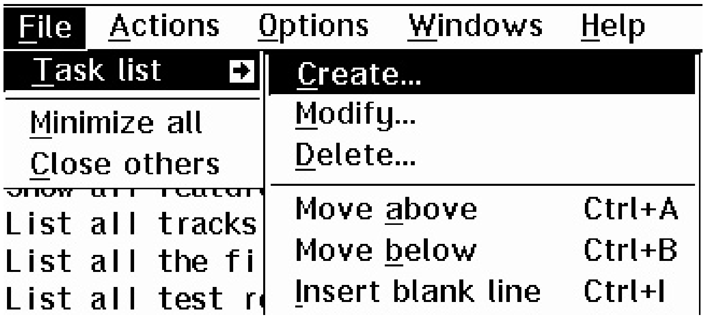


Figure 4. File Menu of CMVC - Tasks Window

The CMVC - Tasks window has the following options available from its File menu in addition to `Minimize all`. They are:

Task list Creates a new task, modifies existing tasks, and deletes tasks from the list. You can also rearrange the entries by moving them up or down the list. You can also add blank lines to act as separators between groups of tasks on the list.

Close others Closes all CMVC GUI windows except the CMVC - Tasks window.

Note: Many of the file menu options display a dialog box when you select the option. You must type the required information in the text entry fields and select a push button to perform the action.

Actions Menu

The Actions menu is on the menu bar of many windows. The menu options represent actions that you can perform on a development object. Related actions are grouped within the menu. Some of the menu options and the actions that you can perform include:

Create... Creates a user ID, a file, release, or task

Open... Opens a defect, or feature

Accept... Indicates that you accept changes proposed by a track

Reject... Indicates that you reject changes proposed by a track

Return... Returns a defect or feature

Delete... Deletes a user ID, a *host list* entry, or a component

Add... Adds a user ID to an access list or other list

Link... Creates a *common file* or a *shared file* within a release

Unlink... Removes common or shared files within a release

Recreate Reinstates a previously defined component

Note: This is not an exhaustive list of actions that you can perform in CMVC. See the various sections that describe the specific actions available to each of the CMVC windows.

Modify Menu

The Modify menu is on the menu bar of many GUI windows. Use the menu options to modify the characteristics of the development object in the current window. Some of the menu options and the characteristics that you can modify include:

Name... Name of a component, level, file, or path

Owner... Owner of a defect, feature, or track

Component... Component associated with a file, release, defect, or feature

Properties... Properties, such as, file mode or level type, that are associated with a file, level, or other CMVC object.

Show Menu

The Show menu is on the menu bar of many GUI windows. Use the menu options either to display information about a development object in the window with which you are working, or to open a related window and display information about a development object that is in the related window. Some of the menu options and the actions that you can perform include:

Details Display the details of a file or other CMVC object

Process configurations Display the shipped and administrator-configured *process* or processes

Defects Open, work with, or modify defects

Features Open, work with, or modify features

Files Work with files that are under CMVC control

Tracks Create, work with, and modify tracks

Levels Create, work with, and modify levels

Releases Create, work with, and modify releases

Host lists Create, work with, or modify host lists

Change history Shows the history of the changes for a defect, feature, file, fix record, level, or level member.

Using the Show Menu

To show information about objects in the window with which you are working, use the following method:

1. Select an item from the window list.
2. Select the corresponding action from the Show menu. For example, you can select `Show details` to display specific information about a list item. The CMVC - Information window is opened and the information that you requested is displayed.

Note: If you do not select an item, a dialog box is opened. Type the required information in the entry fields in the dialog box and select `OK` to display the results.

You can also use the Show menu to access information in windows that are related to the actions that you select. For example, if you choose an item from the window list and select the Show Host lists option in the User Window, the Host Lists window is opened and the requested information is displayed. If you do not select an item, but a default query is defined for the window, the query is issued and the window associated with the query is opened. The requested information is displayed in the client area.

Options Menu

This menu is on the menu bar of all GUI main windows. The menu options and the actions you can perform include:

Family... Selects another CMVC family with which to work.

User ID... Switches to another CMVC user ID to which you have access.

Directory... Specifies the name of the directory in which you want to put your project files when you check files out of the server. The default is the directory where you start the CMVC GUI. The `C:\CMVC\WORK` directory is an example of a working directory.

Print command... Specifies the *command* that is used to access your printer. If no value is specified for this command, the print command is initialized to **PRINT.COM**.

Log file name... Specifies the name of the log file to which commands are logged. If no value is specified for this file, the log file name is initialized to **CMVC.LOG**.

Compare command... Specifies the *compare* command to use in the Show File Differences dialog box of the CMVC - File window. If no value is specified for this option, the compare command is initialized to **DCOMP.EXE**.

Other options... Displays a dialog box containing two check boxes:

- **Set verbose mode**
  Indicate whether you want to see a message upon successful completion of CMVC options. If no value is set for this option, then verbose mode is set to **False**.
- **Set automatic refresh**
  Indicate whether the client area of the main CMVC windows is to be refreshed automatically. If no value is set for this option, then automatic refresh is set to **False**.

Note: Each setting is saved and reused until you change it.

Windows Menu

This menu is on the menu bar of every CMVC main window. Use this menu to navigate the GUI windows.

When you select this menu, the following options are displayed:

- Tasks
- Defects
- Features
- Files
- Tracks
- Levels
- Releases

- Components
- Users
- Change history.

Note: The option that points to the current main window from which the Windows menu is called, is grayed out.

Cascaded menus are provided for the Defects, Features, Tracks, Levels, Releases, Components, and Users windows. Use the cascaded menus to access these windows and their associated windows, such as, the Verification records window or the Approval records window.

Help Menu

There is a Help menu in the menu bar of all GUI windows. Use this menu to get the following help information:

Help index Provides an index to all the topics for which Help is available. Use the index to find and select a topic, and display the information of the selected topic.

General help Displays information about the window with which you are working.

Using help Provides information on how to use the Help facility.

Keys help Displays information about the use of function keys, mnemonics, and shortcut keys in the GUI.

CMVC commands help
Provides information on CMVC commands, their syntax, and examples.

Product information
Provides information about the name and version of the CMVC GUI.

Other Menus

In addition to the File, Actions, Options, and Help menus, a Corequisite menu is provided in the Tracks window. Use this menu to add and remove *corequisite tracks*.

Description of Dialog Boxes

Various types of dialog boxes are displayed when you use CMVC. Some of them require that you type information before continuing with an action whereas other boxes convey messages to you. The CMVC dialog boxes and the information that you can display include:

Help Dialog Box

Displays online help information when you select a help action.

Error Dialog Box

Displays an error number and message when an error occurs. You must close the error dialog box before you can perform any further actions.

Confirmation Dialog Box

Displays a message that requests you to confirm that you want to continue with an action. Select the Yes push button to proceed or select the No push button to stop the action.

Prompt Dialog Box

Prompts you to enter additional information. Type the information in the entry field or fields.

Open List Dialog Box

Displays shipped and administrator-configured entry fields that you can use to type names, comments, and other specific information. These boxes also contain lists, buttons, and a command box that you can use to make selections and start actions. To access these dialog boxes, select Open list... from the File menu in any main GUI window.

Choices Dialog Box

Displays a list of available options for path names and other CMVC objects. To open this dialog box, select the Choices... push button that is next to a text entry field in an open list or prompt dialog box. To fill the entry field, select an item from the list of choices.

Remarks Editor Dialog Box

Allows you to enter your remarks or comments about CMVC objects. To open this dialog box, select the Edit... push button located next to the Remarks field in a dialog box. The box contains a large, scrollable area that you can use to type up to 15 999 alphanumeric characters.

You can either get text from a file or type text directly in the Remarks Editor dialog box.

To get text from a file:

1. Type the name of the file that contains the text of your remarks in the Get remarks from file entry field.
2. Select Get to retrieve the file. The text is displayed in the client area of the Remarks Editor dialog box. Use the scroll bars to view text that is not visible in the client area, or use the Page Up and Page Down keys to scroll through the text.

   Note: Any text that exists in the Remarks Editor dialog box is replaced by the new text from the get file.
3. Add or delete text, as required.
4. Select the OK push button to store the text.

To type text in the Remarks Editor dialog box:

1. Type your remarks on any line in the text area.
2. Press the Enter key when you reach the end of the line. The pointer is repositioned at the start of the next line.
3. Type any additional text.

4. Use the scroll bars to view text that is not visible in the client area, or use the `Page Up` and `Page Down` keys to scroll through the text.
5. Add or delete text, as required.
6. Select `OK` to store the text.

Note: If you press the `Esc` key or the `Cancel` push button, a confirmation message is displayed. This prevents you from losing any information accidentally.

The Parts of a Dialog Box

The parts of a dialog box and their functions are:

Title Describes the information that you must type in a dialog box or the action that you can perform.

Text Entry Label Describes the information required in a text entry field.

Text Entry Field Allows you to type and edit information, such as, a name, a word, a string, or a sentence. To enter information, you can either type text from the keyboard, or select an item from the client area and perform an action. When the corresponding dialog box is displayed, some entry fields may be prefilled with information. Accept the information in the prefilled fields or type new information, as required.

Radio Button Selects a choice from a fixed set of options. The text beside this round-shaped button indicates the option. You can only select one radio button from a set of options.

Drop Down List Combo Box Selects an action from a list of valid actions. The list is normally collapsed into an icon next to the field. You can only select from the choices available in the drop down list. Examples are the `in` and `No Sort` drop down list combo boxes in the Open list dialog box.

Toggle Button Selects or deselects an action. The label above this square-shaped button describes the corresponding action. If several toggle buttons are provided, you can select or deselect any combination of actions.

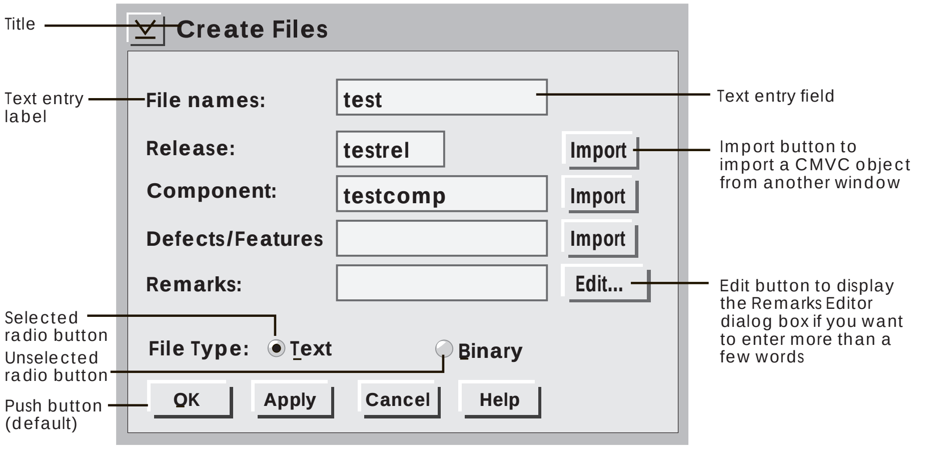
Push Button Starts an action. The text label on this button describes the corresponding action.

Command Box Contains a separate, scrollable area and a text entry field.

The scrollable area displays a history list of the queries you issued for the dialog box during the current working session. To reissue a command, position the cursor on the query and double-click the left mouse button.

The text entry area can be used either to type a new query or modify a query in the history list. To modify a query, select an item from the list of items in the client area. The query is displayed in the entry field and you can add or delete text, as required.

Figure 5 on page 11 and Figure 6 on page 11 show examples of dialog boxes and indicates their parts.



Create Files

File names: test

Release: testrel

Component: testcomp

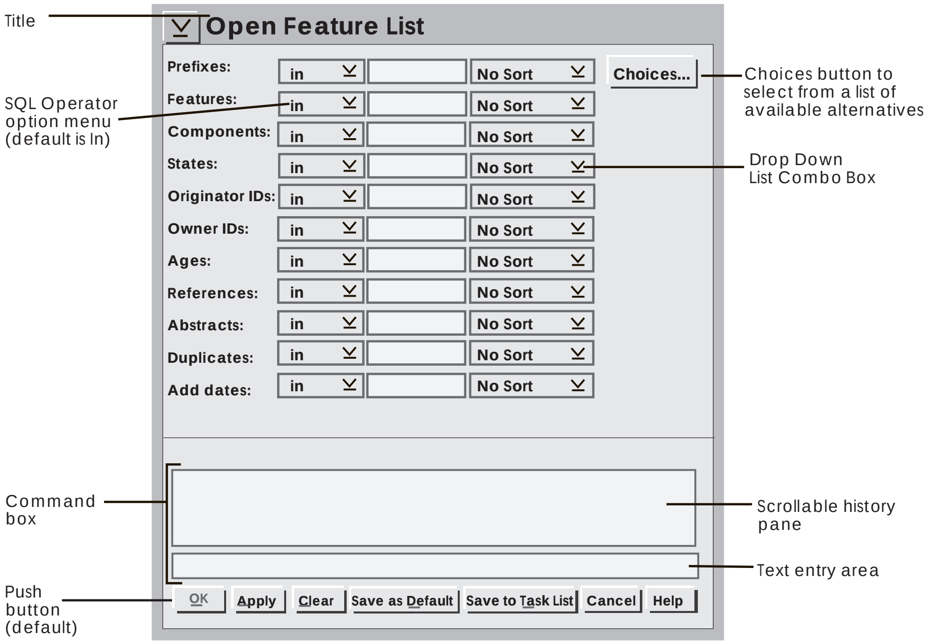
Defects/Features

Remarks:

File Type: ● Text ○ Binary

Title
Text entry label
Text entry field
Import
Import button to import a CMVC object from another window
Import
Import
Edit...
Edit button to display the Remarks Editor dialog box if you want to enter more than a few words
Selected radio button
Unselected radio button
Push button (default)
OK
Apply
Cancel
Help

Figure 5. Parts of Dialog Boxes (1)



Open Feature List

Prefixes:	in		No Sort
Features:	in		No Sort
Components:	in		No Sort
States:	in		No Sort
Originator IDs:	in		No Sort
Owner IDs:	in		No Sort
Ages:	in		No Sort
References:	in		No Sort
Abstracts:	in		No Sort
Duplicates:	in		No Sort
Add dates:	in		No Sort

Figure 6. Parts of Dialog Boxes (2)

SQL Operator (in) and No Sort Combo Boxes

The SQL operator (`in`) and `No Sort` drop down list combo boxes are located beside the text entry fields in Open List dialog boxes. Use these combo boxes to specify the criteria used to perform a query. You can specify criteria for any number of available fields. The *default* for each field is represented by the label on the combo box.

SQL Operator (in) Combo Box

This drop down list combo box contains a list of SQL *operators* that you can use to define an SQL query and perform a *search*.

To use this combo box, select one of the `in` combo boxes that are beside the text entry fields in the Open List dialog box. When you select this combo box, a list of available SQL operators is displayed. Select one of the following operators:

in Searches using a specified list of items.

not in Searches using items not in a specified list of items.

like Searches using a specified string. You can use SQL wildcards with this operator. A percent sign (`%`) represents zero or more characters in a string and an underscore (_) represents one character in a string.

= Searches using a single item.

!= Searches using items not equal to the specified item.

> Searches using alphanumeric strings greater than the specified item.

< Searches using alphanumeric strings less than the specified item.

between Searches for items that are between two specified items.

is null Searches using fields that are set to null or using fields for which no values are assigned. Null differs from the values of zero (`0`) and blank ( ).

Note: To specify a search priority and further structure the query, use the `No Sort` combo box.

No Sort Combo Box

You can apply sorting criteria to the entry fields in an Open List dialog box with the `No Sort` drop down list combo box.

To specify a search priority:

1. Select one of the `No Sort` combo boxes that are beside the text entry fields in the Open List dialog box.
2. Select a priority for the field. The priorities are `First`, `Second`, and `Third`.
3. Start the search. The results are ordered according to the priorities you set for each field.

Note: The First, Second, and Third priorities can only be assigned to one field within a dialog box at a time. You can, however, perform nested sorts by specifying different priority levels for different fields.

Push Buttons

Two or more of the following push buttons are at the bottom of any dialog box:

Apply Performs the action associated with the dialog box but does not close the dialog box.

Cancel Closes the dialog box without performing the action associated with the dialog box.

Clear Clears all text entry fields and returns buttons to the default setting.

Help Provides online help for the dialog box.

No Indicates that you disagree with the question displayed in the dialog box.

OK Performs the action associated with the dialog box and, if no error occurs, closes the dialog box. If the action is not associated with a dialog box (for example, message dialog boxes), this button simply closes the dialog box.

Print Prints the contents of the dialog box. (The print command is either the **print.com** default or the print command that you specified with the `Print command...` option.)

Other push buttons that are next to text entry fields help you enter information effectively. These buttons, and their functions, are:

Choices... Displays a list of items from which you can select. When you select an item, the item is displayed in the corresponding text entry field. The choices in the selection list can be shipped values or can be configured by your *family administrator*. The shipped values are listed in the *IBM CMVC User's Reference*.

Edit... Opens the Remarks Editor dialog box. This dialog box supports multiple line text entry. For additional information about how to use this box, refer to “Remarks Editor Dialog Box” on page 9.

Import Imports one or all of the selected items from a matching main window. When you select an item, it is displayed in the corresponding text entry field.

Some examples of the types of information you can *import* are a selected user ID from the client area of the CMVC - Users window, a selected component from the client area of the CMVC - Components window, or a selected release from the client area of the CMVC - Releases window.

Get Places the contents of a file into the Remarks Editor dialog box when you type the name of an existing file in the adjoining text entry field.

Note: Getting a file will replace the current contents in the Remarks Editor dialog box.

Chapter 2. Using the GUI Windows

This section describes how to perform common tasks with the GUI windows. These tasks include:

- Opening GUI windows
- Displaying information
- Selecting from scrollable lists in GUI windows
- Navigating and performing actions
- Defining default queries and tasks
- Creating reports
- Working with configurable fields and processes
- Showing the change history of CMVC objects.

In addition, this section describes how to use specific CMVC windows to display and work with information in the CMVC - Change History and CMVC - Information windows.

Opening GUI Windows

To open the CMVC main windows, you can:

- Select one of the windows from the Windows pull-down menu in the CMVC - Tasks window.
- Select one of the windows from the Windows popup menu in any GUI window.
- Double-click on a task listed in the client area of the CMVC - Tasks window.
- Select a task listed in the client area of the CMVC - Tasks window. Then select `Execute` from the Actions menu.

Displaying Information in the GUI Client Area

When you issue an SQL query for information, the information you requested is displayed in the client area of the window with which you are working.

To list information in the client area:

- Issue the default query you defined using the `Default list` option from the File pull-down menu. You can create this default list by using the `Open list...` option or the `Default query` option from File pull-down menu. For more information on defining default queries, see "Defining Default Queries and Tasks" on page 19.
- Double-click on a task that is listed in the client area of the CMVC - Tasks window.
- Select a task listed in the client area of the CMVC - Tasks window. Then select `Execute` from the Actions menu. Select `Open list...` from the File menu in the window and use the corresponding dialog box to define a list.

After the list is displayed, you can select one or more items from this list and use the Actions, Modify, and Show menus to perform additional actions on the items.

Selecting from Scrollable Lists in GUI Windows

To increase the speed with which you perform CMVC actions, you can define a query that displays a list of objects in the client area of the window with which you are working. You then use the object-action selection model to work with these objects. This section briefly describes how you can select items from the various types of lists.

Selecting from Lists in the CMVC - Tasks Window

The Tasks list is a list of objects in the CMVC -Tasks window that you can use to perform common actions in your daily development environment.

To select an item from the Tasks list, position the pointer on the line that contains the predefined task that you want to perform.

Note: For more information about tasks, refer to “Defining Tasks” on page 20.

Selecting from Lists in Other GUI Windows

Unlike the CMVC - Tasks window which contains predefined tasks, the other GUI windows contain lists of information about specific CMVC objects, such as, defects, components, releases, or tracks. Each line in these lists contain information about one object. You can select one or more of the objects on which to perform an action.

Selecting a Single Item

To select and perform an action on a single item:

1. To select the object, position the pointer anywhere on the line that contains the item on which you want to perform an action, and click once on mouse button 1.
2. Select the action that you want to perform from the appropriate menu.

Selecting Several Consecutive Items

To select and perform an action on several items:

1. Position the pointer on the first item, click and hold mouse button 1, and drag the pointer until all the items on which you want to perform the action are selected.
2. Select the action that you want to perform from the appropriate menu.

Selecting Many Consecutive Items

To select and perform an action on many items:

1. Position the pointer on the first item and click on mouse button 1.
2. Use the scroll bar to move the pointer until all the items on which you want to perform the action are selected.
3. Press and hold the `Shift` key.
4. Position the pointer on the last item and click on mouse button 1.
5. Release the `Shift` key.
6. Select the action that you want to perform from the appropriate menu.

Modifying an Existing Selection of Items

You can modify an existing selection consisting of one or more items. Modify a selection by adding more items to the existing selection or deselecting items from the existing selection.

Adding Items to a Selection

To add items to a selection:

1. Press and hold the Ctrl key.
2. To select a single item, position the pointer on the item to be added and click on mouse button 1.

   You can add several or many items by clicking and holding mouse button 1, and then dragging the pointer over the items that you want to select.
3. Release the Ctrl key.

Removing Items from a Selection

To remove items from a selection:

1. Press and hold the Ctrl key.
2. To deselect the object, position the pointer on the object and click on mouse button 1.

   You can deselect several or many items by dragging the pointer over the items that you want to deselect.
3. Release the Ctrl key.

Navigating and Performing Actions

This section describes how to navigate the main windows in the GUI, how to navigate scrollable lists and dialog boxes, and how to perform CMVC actions.

If you are using a mouse, you can use the Windows pulldown menu and the Windows popup menu to access the CMVC menus and menu options. If you are using a keyboard, you can use mnemonics to navigate in scrollable lists and dialog boxes.

You can also use mnemonics to select menu bar options. Shortcut keys provide short cuts in performing certain actions because they let you bypass menu item selection.

For additional information about the overall navigation model that applies in your environment, refer to the related books on the OS/2 operating system.

Using the Windows Menus to Navigate GUI Windows

Two Windows menus are provided to help you effectively navigate the window hierarchy. You can access the Windows pulldown menu from the menu bar of all CMVC main windows, or you can access the Windows popup menu from any CMVC window.

When you use the Windows pulldown menu, you must first drag the pointer to select the GUI window that you want to access. If the window is on a cascaded menu, select the window name from the list of menu options. After you make your selection, the window is displayed.

To use the Windows popup menu:

1. Position the pointer anywhere in the client area of the GUI window with which you are working.
2. Press mouse button 2. The popup menu is displayed.
3. Select the window with which you want to work from the popup menu or the cascade menus. The selected window is displayed.

Using the Keyboard to Navigate in Scrollable Lists

If you do not want to use the mouse pointer to navigate within a window or dialog box, you can use the following keyboard alternatives:

↓ Moves the cursor from item to item in the list, moving from the top down

↑ Moves the cursor from item to item in the list, moving from the bottom up

Space Selects the item identified by the location cursor

Enter Starts the default action

Page Down Moves the cursor down through the contents of the window, screen by screen.

Page Up Moves the cursor up through the contents of the window, screen by screen.

Using the Keyboard to Navigate in Dialog Boxes

If you do not want to use the mouse pointer to navigate within a dialog box, you can use the following keyboard alternatives:

↓↑ Moves the cursor from item to item in an up and down direction. For example, use the arrows to move the cursor up and down the options on the popup menus and menus.

←→ Moves the cursor from item to item in a left and right direction. For example, use the arrows to move left and right from push button to push button.

Tab Moves the cursor from field to field in a left-to-right and up-to-down direction. This keystroke does not apply to the Remarks Editor dialog box.

Ctrl Tab Is the same as Tab.

Shift Tab Moves the cursor from field to field, in a right-to-left and down-to-up direction. These keystrokes do not apply to the Remarks Editor dialog box.

Ctrl Shift Tab Is the same as Shift Tab.

Space Selects a field, activates a toggle button, or displays the options menu from the Option button.

Enter Activates the default push button.

Esc Closes the dialog box without performing the action associated with the dialog box.

Using Mnemonics to Select Menu Bar Items

A mnemonic is an underlined character on a menu or menu item that, when entered from the keyboard, starts the action identified by the menu item. Some examples of menu bar mnemonics are:

```
File
Actions
Windows
Options
```

To access the File menu in a main window using mnemonics:

1. Press the `Alt` key and the letter `F` at the same time. The File menu is displayed. It contains the following items, each of which has a unique mnemonic:

   Minimize all

   Close others

2. Type the mnemonic to start the corresponding action.

Using Shortcut Keys to Perform Actions

Certain keys and combinations of keys help you to quickly perform frequently used actions. The CMVC GUI shortcut keys are:

Alt + F4 Closes the window with which you are working

Ctrl + F Finds the next matching item

Ctrl + O Displays the Open List dialog box for the window with which you are working

Ctrl + P Prints the contents of a window

Ctrl + S Saves the current list of objects in a file

Shift + F10 Displays the Windows popup menu from the client area of a GUI window

F5 Refreshes the query for the window and displays the current query results

F10 Moves the focus to the menu bar

Shift + Alt + L Displays the Check Out Files dialog box from the Files window

Shift + Alt + N Displays the Check In Files dialog box from the Files window.

Defining Default Queries and Tasks

You can customize CMVC to your development environment by specifying default queries and tasks for the GUI windows with which you work most frequently. This section briefly discusses how to define default queries and tasks.

Defining Default Queries

You can define a default query in any CMVC window. The query is issued each time the GUI window is opened and the information that you requested is displayed in the client area of the window.

To define a default query for a window:

1. Select `Open list...` from the File menu in the appropriate window. The corresponding Open List dialog box is displayed.
2. Enter the search criteria, with any SQL operators to refine the query, in one or more entry fields, or select a previously executed query from the history pane in the command box, or type a valid SQL query in the entry field in the command box.
3. Select `Apply` to perform the query and review the results.
4. Select `Save as Default` from the Open List dialog box. The query that you executed is saved as the default query.

Note: To modify the query, select `Modify...` from the Default Query option on the File menu.
To delete the query, select `Delete...` from the Default Query option on the File menu.

Defining Tasks

Tasks are displayed in the client area of the CMVC - Tasks window. There are two ways to define a task, depending on whether you are defining a task in the CMVC - Tasks window or in another GUI window.

CMVC - Tasks Window

To define a task in the CMVC - Tasks window:

1. Select `Task list` from the File pull-down menu. A cascaded menu appears.
2. Select `Create...` from the cascaded menu. The Create Tasks dialog box is displayed.
3. Type a title for the task in the `Description` field. This title is displayed in the task list each time that you open the CMVC - Tasks window.
4. Type a valid SQL query, or the name of any executable command, *shell script*, or program in the `Task` field.
5. Select either the `Query` or `Executable` push button to indicate whether the task is an SQL query, or an executable command, shell script, or program.
6. If the task is a query, select the window to which the task applies from the list of windows in the Window list box.
7. Select `OK` to create the task. The task is inserted in the Tasks list in the CMVC - Tasks window below the currently selected task.

Note: When you issue an SQL query, the corresponding window opens and query results are displayed in the client area of the window. When you start an executable task, program results are displayed in the CMVC - Information window.

You can also modify, delete, or reorder tasks in the Tasks list by selecting `Modify...`, `Delete...`, `Move above`, `Move below`, or `Insert blank line` from the `Task list` cascaded menu.

Other GUI Windows

To define a task in any other GUI window:

1. Select `Open list...` from the File menu. The corresponding Open List dialog box is displayed.
2. Enter the search criteria with any SQL operators to refine the query in one or more entry fields or select a previously defined query from the history pane in the command list, or type a valid SQL query in the text entry field in the command box.
3. Select `Apply` to perform the query and verify the results.
4. Select `Save to to Task list...` from the Open List dialog box. The Save to Task List dialog box is displayed.
5. Type the task title in the `Task title` field. The Window and Query fields are prefilled with the task that you specified.
6. Select `Save` to save the task. The task is added to the Tasks list in the CMVC - Tasks window below the selected task.

Creating Reports

You can use the Open List dialog box to create a report. When you define and execute a query, the report information is displayed in the corresponding window. You can then print or save the report to a file.

The report formats consist of:

Window list Creates a report consisting of the contents of the window, such as a defect list

Report table Creates a report table consisting of the formatted output of selected table entries from a table stored in the CMVC database

Report long Creates a long report consisting of a detailed list of information for selected items stored in the CMVC database

Certain report formats can only be used in specific windows. For example, you can specify the Window list, Report table, and Report long formats for the Track, Level, User, Component, Defect, and Feature windows. You can only specify the Window list and Report table formats for the other GUI windows.

To create a report and save the window contents to a file:

1. Use the Open List dialog box to define and execute a query. The report information is displayed in the client area of the window. For more information about the Open List dialog box, refer to “File Menu” on page 3.
2. To save the report, select `Save as...` from the File menu on the window menu bar. The Save List to File dialog box appears.
3. Type a file name in the Output file field.

   Note: If you do not specify an *absolute path name*, the file is saved relative to the directory that you set with the `Directory` option.

4. Select the output format in the Save List to File dialog box. Valid formats are window list, report table, or report long.
5. Select `OK` to save the report.

To create a report and print the window contents:

1. Use the Open List dialog box to define and execute a query. The report information is displayed in the client area of the window.
2. Select `Print...` from the File menu on the window menu bar. The Print List dialog box is displayed.
3. Select the output format. Valid formats are window list, report table, or report long.
4. Select `OK` to print the report.

You can also create reports by defining tasks that issue the **Report** command. For more information about the **Report** command, refer to the book *IBM CMVC Commands Reference*.

Working with Configurable Fields

Depending on your organization's requirements, your administrator can add fields, suppress certain shipped fields, or rename certain shipped fields. Your family administrator can also configure up to 20 additional fields, containing as many as 85 alphanumeric characters, for use with the defect, feature, file, and user objects. These fields are configured for the `open` and `modify` actions only.

You can use the text entry fields in CMVC dialog boxes to specify a wide range of search criteria and other information when you issue queries or perform CMVC actions. You can work with both the shipped fields and the configured fields that your family administrator customizes to your development environment. You can use configured fields, for example, to store phone numbers, phase-injected data, or other relevant information.

When discussing the `open` and `modify` actions in the chapters that deal with certain CMVC objects, this book provides a complete description of all shipped fields. Fields can be configured for the following GUI windows:

- Defects
- Features
- Files
- Users.

If your family administrator has added fields, suppressed fields, or renamed fields, however, the fields that are displayed in certain dialog boxes may differ from those described in this book. For a summary of the configured fields in your environment and the information that you can enter, see your family administrator. For a list of the shipped fields, refer to the book *IBM CMVC Server Administration and Installation*.

Working with Configurable Processes

In addition to configuring fields, another way that your family administrator can customize CMVC to your development environment is by configuring the processes that apply to a release or component. Your family administrator can configure a process grouping one or more CMVC subprocesses.

When processes are configured, the states that an object moves through vary depending on the CMVC subprocesses that are included in the process for the associated release. For example, if the test *subprocess* is not included in the process for a release, the test state of a track is bypassed. If the level subprocess is not included in the process for a release, the integrate state of a track is bypassed.

For a list of the processes that are available and the subprocesses that are included in each process for specific releases or components, see your family administrator. To display specific process configurations for defects, features, releases, and components, select `Process configurations` from the Show menu in the corresponding window. For more information about configurable processes, refer to the books *IBM CMVC Server Administration and Installation* and *IBM CMVC Concepts*.

Using the CMVC - Change History Window

You can show the history of changes to certain CMVC objects, including:

- Defects
- Features
- Files
- Levels
- Tracks
- Level members
- Fix records.

There are two ways to display the CMVC - Change History window. You can either select `Change history` from the Show menu in the corresponding GUI window, or you can select `Open list...` from the File menu in the Change History window.

Using the Show Menu

To show the change history for a CMVC object with the Show menu:

- Select the defect or feature for which you want to display the history of changes from the list of defects or features in the client area of the Defects, Features, Tracks, Levels, Level Members, or Fix Records windows. You can also select a file from the client area of the Files window.
- Select `Change history` from the Show menu of the window with which you are working. The CMVC - Change History window is opened and the change history information is displayed in the client area.

Using the Change History Window

To show the change history for a CMVC object with the Change History window:

1. Select `Open list...` from the File menu in the Change History window. The Open Change History List dialog box is displayed.
2. Enter the search criteria, along with any SQL operators to refine the query, in the dialog box.
3. Select `OK` to display the change history. The window is refreshed and the change history of the CMVC object is displayed in the client area.

Using the CMVC - Information Window

When you request information or perform an action with the Show menu in certain GUI windows, the results are displayed in the CMVC - Information window. Use the CMVC - Information window to review, save, or print the results, or to search for a selected text string.

Use the various options from the File pull-down menu of the CMVC - Information window for the following actions:

Option	Purpose
Save	Stores the listed information to an output file.
Save as...	Stores the listed information to an output file that you name. You enter the file name in a Save As dialog box that appears.
Find...	Searches for a text string in a list of text items. You enter the text string in the Find Text in List dialog box that appears.
Find next	Searches for the next occurrence of a specified text.
Print	Prints the information listed on the window.

Chapter 3. Working with Files

You can use the CMVC - Files window to bring files under CMVC control and to perform a variety of file management tasks on these files. This chapter describes how to display information in the client area of the CMVC - Files window, and how to perform queries and other common file management actions. If you have superuser authority, you can perform any of the actions described in this chapter.

File Management Actions

The range of actions that you can perform depends on your level of authority. For example, if you are the owner of a component, you have *implicit authority* to perform all CMVC file management actions on the files managed by the component. You can also perform specific file management actions if you have one of the following authority types:

- Explicit authority for the component associated with the file.
- Explicit authority for a parent of the component associated with the file. You can inherit authority for the *child component* provided that the component associated with the files does not have *restricted authority*.

Some of the file management actions that you can perform with the CMVC - Files window include:

- Creating, checking in, checking out, extracting, locking, and unlocking files on the CMVC server
- Linking (creating common or shared files), unlinking, and re-creating deleted files
- Deleting, destroying, and modifying path names, components, and file modes
- Showing the file details, file differences, and user IDs that are associated with locked files
- Showing the file change history.

Displaying File Information in the Client Area

When you use the CMVC - Files window to issue a query, specific file information is displayed in the client area of the window under the following column headings:

Path Name Displays the *path name* for the file that is under CMVC control. This can consist of directory names and a base name.

Release Displays the name of the release with which the file is associated.

Component Displays the name of the component that manages the file.

Current Displays the current version identifier for the file.

Committed Displays the most recent version identifier for a file that belongs to a committed level.

There are two exceptions. If the tracking subprocess is included in the process for the release, the current version identifier is displayed. If the level subprocess is not included in the process for the release, the version identifier associated with the most recently committed track is displayed.

Locked by Displays the user ID of the person who has checked out or locked the file.

Mode Displays the file mode. The mode is a 4-digit octal number that indicates the file permission.

Created Displays the date and time (yy/mm/dd hh:mm:ss) that the file was placed under CMVC control.

Deleted Displays the date and time (yy/mm/dd hh:mm:ss) that the file was deleted.

Updated Displays the date and time (yy/mm/dd hh:mm:ss) that the file was last changed.

Note: If your family administrator has configured extra fields, additional columns may be displayed when you issue a query for file information. For more information about the configured fields that are used in your environment, see your family administrator.

Performing Queries with the Open File List Dialog Box

The Open File List dialog box allows you to perform queries and list information in the CMVC - Files window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to "Defining Default Queries and Tasks" on page 19.

To perform a query using the Open List dialog box:

1. Select `Open list...` from the File menu to display the Open File List dialog box.
2. Type the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

   Path names Path names for the files that are under CMVC control. For example, you may want to list all the files that have the path name `test/view/_x.c`.

   Base names Base names for the files that are under CMVC control. For example, you may want to list all your `Makefile` files, regardless of the directory names with which they are associated.

   Releases Name of the release to which the files belong. For example, you may want to list all of the files associated with the `debugger4` release.

   Components Name of the components to which the files belong. For example, you may want to list all files that are managed by the `tools` component.

   Current versions Current versions of the files. For example, you may want to list all files whose current version is `2.2`.

Committed versions Most recent versions of files that belong to a committed level. For example, you may want to list all files in committed version 2.2.

There are two situations in which the most recent version is not used. If the tracking subprocess is not included in the process for the release, the current version identifier is displayed. If the level subprocess is not included in the process for the release, the version identifier associated with the most recently committed track is displayed.

Locked by User ID of the person who checked out or locked the file.

Mode A 4-digit octal number that indicates the file permission.

Add dates Date (yy/mm/dd) that the files were placed under CMVC control. For example, you may want to list only those files that were created after 93/12/29. Valid SQL operators for this field are <, >, like, is null, and between.

Delete dates Date (yy/mm/dd) that the files were removed from CMVC control. For example, you may want to list those files that were deleted after 93/12/29. Valid SQL operators for this field are <, >, like, is null, and between.

Change dates Date (yy/mm/dd) that the files were last changed. For example, you may want to list only the files that were changed after 93/12/29. Valid SQL operators for this field are <, >, like, is null, and between.

3. Type the information required in any additional fields that your family administrator has configured.

   Note: You can also select a previously defined SQL query from the history list in the Command box or type an SQL query in the entry field in the Command box.

4. Select OK to perform the query. The CMVC - Files window is refreshed, and the information that you requested is displayed in the client area.

You can use the Refresh now item on the File menu to reissue a query to the server. The most recent query is sent and the list is updated with current information from the server. If you issue a query to the server using the Open list... option, you can replace the query with the default query specified in the system file by selecting the Default list item from the File menu.

Note: For more information about how to use the Options menu items to specify operators, refer to “SQL Operator (in) and No Sort Combo Boxes” on page 11.

Performing Actions with the CMVC - Files Window

This section describes the authority and prerequisites for performing common file management actions and how to perform these actions.

Creating a File

You can take an existing file from your workstation and place it in CMVC control. To do this, you must associate the file with a release (to relate the file to a development effort) and with a component (to control the ownership of and access to the file).

Authority Required: The component owner has implicit authority to create files. You have *explicit authority* if the FileAdd action is specified for your user ID in the component that manages the file.

Prerequisites: When creating files, you must ensure that the following prerequisites are met:

- The text or binary file must already exist in your workstation environment.
- The file must have a unique name in the specified release with which it is associated.
- You must have read/write permission for the file and the directory in which the file is located.

Procedure: To create a file:

1. Select `Create...` from the Actions menu on the CMVC - Files window menu bar.
2. Type information to specify the name and location of the file you want to place under CMVC control in the following fields:

   File names Name of the file that you want to create. Directory names and *base file names* can be included. Examples are `source/view.c` or `view.c`. You cannot use blanks, tabs, or control characters in this field.

   Note: Pathnames can be specified using the OS/2 convention as well. An example is `source\view`.

   Release Name of the release to which files are assigned. You can also import a selected release from the CMVC - Releases window.

   Component Name of the component that will manage the files. You can also import a selected component from the CMVC - Components window.
3. If you are creating a file for which the tracking subprocess is included in the process for the release, you must type the identifiers of the defects or features associated with the tracks in the `Defects/Features` field. You can also import selected defects or features from the CMVC - Defects or CMVC - Features window.

   Note: The tracks must be in the `fix` *state* and any *fix record* that exists for those tracks must be in the `ready` or `active` state.
4. If you want to type additional comments, select `Edit...`. The Remarks Editor dialog box is displayed. You can type up to 15 999 alphanumeric characters in this box. An example is:

```
The files view.a view.b and view.c were placed under
CMVC control, referencing the debugger release and
the debugadmin component.
```

5. Select either the `Text` or `Binary` radio button to indicate the type of file you are placing under CMVC control. The default is to create a text file.

6. Select `OK` to create the files.

The files are created in CMVC with the same file mode (or file permissions) that they have on the client workstation. (The original file remains as a copy on your workstation, with read-only permission.) If you want to change the file permission that CMVC uses when checking in or extracting files, modify the file mode of the files on the workstation before placing them under CMVC control.

Version Control System Considerations: If CMVC uses the Source Code Control System (SCCS) as the underlying version control system, any text files that have one or more of the following characteristics are created as binary files:

- ASCII control character `SOH` at the beginning of a line
- ASCII control character `NUL` anywhere in the file.

The contents of the file are not affected. There are no restrictions on text files if CMVC uses the PVCS Version Manager** system as the version control system.

Checking Out a File

You can *check out* the current version of a file to a client workstation so that you can make changes to the file. For more information about extracting copies of current or earlier versions of files, see “Extracting a Copy of a File” on page 31.

Authority Required: The component owner has implicit authority to perform multiple check outs on common files. You have explicit authority if the FileCheckOut action is specified for your user ID in the component associated with the file.

The component owner has implicit authority to check out a file that is already locked in another release. You have explicit authority if the FileForceOut action is specified for your user ID in the component associated with the file.

Prerequisites: When checking out files, you must ensure that the following prerequisites are met:

- You must have read/write permission in the destination directory.

- There is enough space in the destination directory.

- If the file exists in the destination directory, the *CMVC client* must have permission to read the file

- File is not marked as *deleted* in CMVC.

Procedure: To check out files from the CMVC server:

1. Select the files that you want to check out from the list of files in the client area.

2. Select `Check out...` from the Actions menu to display the Check Out Files dialog box. Accept the prefilled information in the `Path names` or `Release` fields, or type new information as follows:

Path names Path names for the files that are under CMVC control. These can consist of directory names and the base file names.

Release Name of the release associated with the files that you want to check out.

Destination directory Directory on your workstation in which you want to store the files that you are checking out. This field is initialized with the value you set with the Directory option in the Options menu. You must have read/write permission.

3. If you want to force the check out when a file exists in another release, select `Multiple check out on common files`.
4. Select `OK` to check out the files. A new version of the file is created and the new version becomes the current version of the file when you *check in* the file.

The file is placed in the destination directory with read/write permission for the owner. The group and other file permissions are set to the file mode that CMVC has recorded for the file.

The file is locked on the server so that other users cannot check it out. They can, however, display information about the file, display the file's contents, or *extract* the file.

Note: If you do not want to select files from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Checking In a File

You can check a file into CMVC to make any file changes permanent. You do this after you edit the contents of the file and save the changes on your workstation.

You cannot change the name of the file while it exists on your workstation. You must change the name of the file in CMVC first. For more information about changing the file name, refer to "Renaming a File" on page 38.

Authority Required: The component owner, or the user who checked out or locked the file, has implicit authority. You have explicit authority if the FileCheckIn action is specified for your user ID in the component associated with the file.

The component owner has implicit authority to break the common link while checking in a file. You have explicit authority if the FileForceIn action is specified for your user ID in the component associated with the file.

Prerequisites: When checking in files, you must ensure that the following prerequisites are met:

- File is in a source directory on your workstation with read/write permission
- File is under CMVC control and is locked or checked out by you

Procedure: To check a file into the CMVC server:

1. Select the files that you want to check in from the list of files in the client area.
2. Select `Check in...` from the Actions menu to display the Check In Files dialog box. Accept the prefilled information in the `Path names`, `Release`, or `Source directory` fields, or type new information as follows:

Path names Path names for the files under CMVC control. These can consist of directory names and the base file names.

Release Name of the release associated with the files that you are checking in.

Source directory Directory on your workstation in which the files that you are checking in are located. You must have read/write permission for the source directory. This field is initialized with the value you set using the Directory option in the Options menu.

3. If the tracking subprocess is included in the process for the release, you must associate the files with specific tracks. Type the identifiers of the defects or features in the `Defects/Features` field. You can also import selected defects or features from the CMVC - Defects or CMVC - Features window.
4. If you are checking in a file that is common to other releases and the tracking subprocess is included in the process for the release, you must specify those releases to maintain commonality. To do this, type the name of the releases in the `Common releases` field. You can also import selected releases from the CMVC - Releases window.
5. If you want to type additional comments, select `Edit....` The Remarks Editor dialog box is displayed. You can type up to 15 999 alphanumeric characters in this box. An example is:

   ```
   Four lines of code were added to file view.c.
   ```
6. If required, select `Break common link`. You may want to maintain the commonality of files for the specific releases identified in the `Release` and `Common releases` fields while breaking the common link in all other unspecified releases in which the file is common. The default is to keep the common link.
7. Select `OK` to check in the files. CMVC recognizes the checked-in version of the file as the current version. The permission for the file in the source directory on your workstation is changed. The owner now has read-only permission. The group and other file permissions are not altered.

If you chose to break the common link, the file is no longer a common file in the release in which you break the common link. It becomes a shared file in the releases in which the link was broken and the file branches to another version number.

Note: If you do not want to select files from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Extracting a Copy of a File

You can extract a file to get a copy of any version of a file on your workstation.

Authority Required: The component owner has implicit authority. You have explicit authority if the FileExtract action is specified for your user ID in the component associated with the file.

Prerequisites: When checking out files, you must ensure that the following prerequisites are met:

- You have read/write permission for the destination directory.
- There is sufficient storage space in the destination directory.

- If the file exists in the destination directory, the CMVC client must have read permission for the file.
- The file must not be marked as deleted in CMVC.

Procedure: To extract a file from the CMVC server:

1. Select the file that you want to extract from the list of files in the client area.
2. Select `Extract...` from the Actions menu to display the Extract Files dialog box. Accept the prefilled information in the `Path names`, `Release`, or `Destination directory` fields, or type new information as follows:

   Path names Path names for the files under CMVC control. These can consist of directory names and the base file names.

   Release Name of the release associated with the files that you are extracting from.

   Destination directory
   Directory on your workstation in which you want to store the files that you are extracting. This field is initialized with the value you set with the Directory option in the Options menu.

   Version Specify the version of a file you want to extract if the version is other than the current version. If you leave this field blank, the current version of the file is extracted.

   File permissions Read, write, and execute file permissions for the extracted files in the 4-digit octal number notation. The default is the file mode associated with the CMVC file stored on the server, with read-only access for all users.

3. If you do not want to substitute assigned values in place of keywords, select `Expand keywords`. The default is to substitute values for the keywords. For a list of supported keywords, refer to the book *IBM CMVC User's Reference*.
4. Select `OK` to extract the file. A copy of the file is placed on your workstation in the directory defined by the path name of the CMVC file and the destination.

You can make changes to the copy of the file on your workstation. You cannot check in that copy of the file to save those changes unless you *lock* the file. It is possible, however, to create this copy as a separate file in CMVC. For more information about locking files, refer to "Locking a File."

Note: If you do not want to select files from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Locking a File

You can lock a file so that it is not available for other users to edit or so that you can check in a file from your workstation.

Authority Required: The component owner has implicit authority to lock a file. You have explicit authority if the FileLock action is specified for your user ID in the component associated with the file.

If the FileLockForce action is specified for your user ID in the component associated with the file, you have explicit authority to lock a common file that is already checked out or locked in another release.

Prerequisites: When locking files, you must ensure that the following prerequisites are met:

- File is not checked out or already locked
- File is not marked as deleted in CMVC

Procedure: To lock files in the CMVC server:

1. Select the files that you want to lock from the list of files in the client area.
2. Select `Lock...` from the Actions menu to display the Lock Files dialog box. Accept the prefilled information in the `Path names` or `Release` fields, or type new information.
3. If required, force the lock operation by selecting `Multiple locks on common files`.
4. Select `OK` to lock the files. The files are locked and cannot be checked out until they are unlocked. Other users can still extract a copy or display the contents of any version of the file.

Note: If you do not want to select files from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Unlocking a File

You may want to unlock a file for the following reasons:

- Cancel any changes made to the checked out file on a workstation so that the changes are not submitted to CMVC.
- Enable other users to check out the file from CMVC.

Authority Required: The component owner or the user who locked or checked out the file has implicit authority. You have explicit authority if the FileUnlock action is defined for your user ID in the component associated with the file.

Prerequisites: When unlocking files, you must ensure that the file was locked by a previous lock or checkout operation.

Procedure: To unlock files on CMVC:

1. Select the files that you want to unlock from the list of files in the client area.
2. Select `Unlock...` from the Actions menu to display the Unlock Files dialog box. Accept the prefilled information in the `Path names`, `Release`, or `Source directory` fields, or type new information.
3. If the file you want to unlock exists on your workstation, use the `Source directory` field to type the directory in which the files you are unlocking are located. You can type over any prefilled information.
4. Select `OK` to unlock the files.

If the file that you unlocked also exists on your workstation, the permission for the file is changed. The owner now has read-only permission.

Note: If you do not want to select files from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Creating Common or Shared Files by Linking

If you are working with two or more releases for which tracking has been specified, and you need to use the same version of a file in all of the releases, then you can create a common file. If no tracking has been specified for the release, the files are shared (not common).

Authority Required: The component owner has implicit authority. You have explicit authority if the FileLink action is specified for your user ID in the component associated with the file.

Prerequisites: When linking files, you must ensure that the following prerequisites are met:

- Existing tracks that will monitor this change to the file in the new release and their fix records are in the active or ready state.
- File is not marked as deleted in CMVC.
- If the file already exists in the release to which you are linking and that release is specified, then there must be no changes that are pending (or uncommitted) in the file.

Procedure: To link files, use the following method:

1. Select the files that you want to link from the list of files in the client area.
2. Select `Link...` from the Actions menu to display the Link Files dialog box. Accept the prefilled information in the `Path names` or `Release` fields or type new information.
3. Type the name of the release to which you want to link the files in the `New release` field. You can also import a selected release from the CMVC - Releases window.
4. If you do not want to link the current version of the files, you must type the version you want to use in the `Version` field. The default is to link the current version of the files.
5. If the tracking subprocess is included in the process for the destination release, you must specify the tracks that monitor the changes made to the release in the `Defects/Features` field. You can import a defect or feature from the CMVC - Defects or CMVC - Features window.
6. Select `OK` to link the files.

You can create common files only if tracking is specified for the process of the source and destination releases. Therefore, you must specify the track or tracks associated with this action by indicating the destination release, and the corresponding defect or feature number.

You can create shared files for releases for which tracking has not been specified. In this case, no track is required.

Note: If you do not want to select files from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Undoing Changes to a File

If the tracking subprocess is included in the process for the release, you can undo the following actions:

- Most recent delete, rename, recreate, or check in action in one or more of the releases in which the file is common
- Most recent create or link in a single release
- Multiple check in actions, up to the latest committed version of the specified file.

If the tracking subprocess is not included in the associated process for the release with the file, you can undo multiple check in actions dating to the first version of the file that was placed under CMVC control.

Authority Required: The component owner has implicit authority to undo changes to a file. You have explicit authority if the FileUndo action is specified for your user ID in the component associated with the file.

The component owner has implicit authority to undo the changes to a common file only in specified releases and thereby break the commonality of the file. You have explicit authority if the FileUndoForce action is specified for your user ID in the component associated with the file.

Prerequisites: When undoing changes to a file, you must ensure that the following requirements are met:

- File is not destroyed.
- Existing tracks that indicate the change to the file that you want to undo are in the fixed state and their associated fix records are in the active state.
- Change that you want to undo (as indicated by the track) is the most recent change to the file.

Procedure: To undo changes to files:

1. Select the files for which you want to undo changes from the list of files in the client area.
2. Select `Undo...` from the Actions menu to display the Undo Files dialog box. Accept the prefilled information in the `Path names` and `Release` fields, or type new information as follows:

   Path names Path names for the files under CMVC control. These can consist of directory names and the base file names.

   Release Name of the release associated with the files that you are undoing changes to.

   Defects/Features If the tracking subprocess is included in the process for the release, type the identifiers of the tracks. You can also import selected defects or features from the CMVC - Defects or CMVC - Features window.

   Common releases
   If you want to undo changes in a file that is common to other releases, specify the releases in which the file is common. You can also import selected releases from the CMVC - Releases window.

3. If you are undoing changes to a file that is common to other releases and you want the changes undone only in the file for the specified release, select `Undo only in the specified releases`. The default is not to undo the changes.
4. Select `OK` to undo the changes. The most recent action is undone. If you are undoing a check in, the version numbers are updated.

Note: If you do not want to select files from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Deleting a File

You can delete files that are under CMVC control.

Authority Required: The component owner has implicit authority to delete a file. You have explicit authority if the FileDelete action is specified for your user ID in the component associated with the file.

The component owner has implicit authority to delete a common file only in specified releases. You have explicit authority if the FileDeleteForce action is specified for your user ID in the component associated with the file.

Prerequisites: If the tracking subprocess is included in the process for the release, all track changes to the file must be committed before the file can be deleted. If the tracking subprocess is not included in the process for the release, the file can be deleted at any time.

Procedure: To delete files that are under CMVC control:

1. Select the files that you want to delete from the list of files in the client area.
2. Select `Delete...` from the Actions menu to display the Delete Files dialog box. Accept the prefilled information in the `Path names` and `Release` fields or type new information.

   If the tracking subprocess is included in the process for the release, you must:

   a. Indicate the tracks that are referenced by the files you are deleting. Type the identifiers of the defects or features in the `Defects/Features` field. You can also import selected defects or features from the CMVC - Defects or CMVC - Features window.

   b. If the file is common to other releases and you want to delete the file in these releases, type the release names in the `Common releases` field. You can also import selected releases from the CMVC - Releases window.

   c. If you are deleting a file that is common to other releases and you want the file to be deleted only for the specified releases, select `Delete only in the specified releases`.
3. Select `OK` to delete the files.

A deleted file is marked as deleted in CMVC but it is not removed. Consequently, you can still display a deleted file in a list on the CMVC - Files main window. You cannot use the name of a deleted file when you are creating another file in the same release.

Refer to “Recreating a Deleted File” on page 37 for instructions on how to re-create a deleted file.

Note: If you do not want to select files from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Recreating a Deleted File

You can recreate files that were deleted from CMVC control.

Authority Required: The component owner has implicit authority to re-create a deleted file. You have explicit authority if the FileRecreate action is specified for your user ID in the component associated with the file.

The component owner has implicit authority to recreate a common file only in specified releases. You have explicit authority if the FileRecreaForce action is specified for your user ID in the component associated with the file.

Prerequisites: When re-creating a deleted file, you must ensure that the following requirements are met:

- File is marked as deleted in CMVC
- File is not destroyed.

Procedure: To re-create files that have been deleted:

1. Select the deleted files that you want to re-create from the list of files in the client area.
2. Select `Recreate...` from the Actions menu to display the Recreate Files dialog box. Accept the prefilled information in the `Path names` and `Release` fields, or type new information.

   If the tracking subprocess is included in the process for the release, you must:

   a. Indicate the tracks that are referenced by the files you are re-creating. Type the identifiers of the defects or features in the `Defects/Features` field. You can also import selected defects or features from the CMVC - Defects or CMVC - Features window.

   b. If the file is common to other releases and you want to recreate the file in all of these releases, type the release names in the `Common releases` field. You can also import selected releases from the CMVC - Releases window.

   c. If you are re-creating a file that is common to other releases and you want the file to be re-created only for the specified release, select `Recreate only in the specified release`. The default is to keep the common link.
3. Select `OK` to re-create the files. The file is re-created under CMVC control.

Note: If you do not want to select files from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Destroying a File

By destroying a file under CMVC control, you can reuse the path name for another file that you want to bring under CMVC control.

Authority Required: The component owner has implicit authority. You have explicit authority if the FileDestroy action is specified for your user ID in the component associated with the file.

Prerequisites: You can *destroy* a file without first deleting it if the process for the associated release does not include the tracking subprocess.

Files that are associated with releases that have the tracking subprocess specified can be destroyed only if they were deleted, and if the deletion was committed.

Warning: You cannot undo this action.

Procedure: To destroy a file:

1. Select the file that you want to destroy from the list of files in the client area.
2. Select `Destroy...` from the Actions menu to display the Destroy Files dialog box. Accept the prefilled information in the `Path names` and `Release` fields, or type new information.
3. Select `OK` to destroy the files.

Previously committed levels can reference files that are destroyed. The destroyed file remains in the version control system.

Note: If you do not want to select files from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Renaming a File

You can rename a file in CMVC.

Authority Required: The component owner has implicit authority to rename a file. You have explicit authority if the FileRename action is specified for your user ID in the component associated with the file.

The component owner has explicit authority to rename a common file only in specified releases. You have explicit authority if the FileRenameForce action is specified for your user ID in the component associated with the file.

Prerequisites: When renaming files, you must ensure that the following requirements are met:

- File is not checked out, locked, or deleted.
- If the file is associated with a release, the process for which includes the tracking subprocess, then the file must not have any changes pending or uncommitted.

Procedure: To rename a file:

1. Select the file you want to rename from the list of files in the client area.
2. Select `Path name...` from the Modify menu to display the Modify File Path Name dialog box. Accept the prefilled information in the `Path names` and `Release` fields or type new information.
3. Type in a new file name in the `New name` field.

   If the tracking subprocess is included in the process for the release, you must:

   a. Indicate the tracks that are referenced by the files you are renaming. Type the identifiers of the defects or features in the `Defects/Features` field. You can also import selected defects or features from the CMVC - Defects or CMVC - Features window.

   b. If the file is common to other releases and you want to rename the file in these releases, you must type the names of the releases in the `Common releases` field. You can also import selected releases from the CMVC - Releases window.

   c. If you are renaming a file that is common to other releases, and you want the file to be renamed only for the specified releases, select `Rename only in the specified releases`.

4. Select `OK` to rename the files. The file is renamed.

If the file is associated with a release for which the tracking subprocess is included, the rename is not made permanent until you commit the track. This means that the rename can be undone, if required.

Note: If you do not want to select files from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Modifying the Component That Manages a File

You can assign a file to a different component.

Authority Required: The component owner has implicit authority. You have explicit authority if the FileModify action is specified for your user ID in the component associated with the file.

Prerequisites: The file is not locked or checked out.

Procedure: To modify the component that manages a file:

1. Select the file for which you want to modify the component from the list of files in the client area.
2. Select `Component...` from the Modify menu to display the Modify File Component dialog box. Accept the prefilled information in the `Path names` and `Release` fields or type new information.
3. Type the name of the component that will assume ownership of the file and manage access to the file in the `New component` field. You can also import a selected component from the CMVC - Components window.
4. Select `OK` to modify the component. The file is now controlled by the new component's access and notification lists.

Note: If you do not want to select files from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Modifying the Properties of a File

You can modify certain properties of a file, such as, the file mode.

Authority Required: The file owner has implicit authority. You have explicit authority if the FileModify action is specified for your user ID in the component associated with the feature.

Prerequisites: The file is under CMVC control.

Procedure: To modify the properties of a file:

1. Select the file for which you want to modify properties from the list of files in the client area.

2. Select `Properties...` from the Modify menu to display the Modify File Properties dialog box. Accept the prefilled information in the `Path names` and `Release` fields, or type new information.
3. Type the file mode for the file in the `File Mode` field. The file mode is a 4-digit octal number that indicates the file's read, write, and execute permissions.
4. Select `OK` to modify the properties. Any changes that you make are reflected in the updated file.

Note: If you do not want to select files from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Showing the Contents of a File

You can show the contents of a file.

Authority Required: The component owner has implicit authority. You have explicit authority if the FileExtract action is specified for your user ID in the component associated with the file.

Prerequisites: When showing the contents of files, you must ensure that the following requirements are met:

- Version of the specified file exists.
- Files must be text files. You cannot display the contents of binary files. File is not marked as deleted.

You can view the contents of a file in the CMVC - Information window.

Note: This is the default action performed when you double-click on a file in the client area of the CMVC - Files window.

Procedure: To display information for one or more files:

1. Select the files for which you want to see the contents from the list of files in the client area.
2. Select `Contents...` from the Show menu to display the Show File Contents dialog box. Accept the prefilled information in the `Path names` and `Release` fields.
3. If you want to view the contents of earlier versions of files, specify the version in the `Version` field. The default is to display the current version of the files.
4. If you do not want to substitute assigned values in place of keywords, select `Expand keywords`. The default is to substitute values for the keywords. For a list of supported keywords, refer to the book *IBM CMVC User's Reference*.
5. Select `OK` to show the contents of the files. The CMVC - Information window is opened and a copy of the file is displayed.

The file is not locked on the server; other users can check it out. The file is not extracted to your workstation.

Note: If you do not want to select files from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Showing the Details of a File

You can view information about files, including:

- Date of creation and last modification
- Indication of file locked status
- List of common files
- List of file versions and their creation dates.

Authority Required: The component owner has implicit authority. You have explicit authority if the FileView action is specified for your user ID in the component associated with the file.

Prerequisites: None.

Procedure: To show information about files:

1. Select the files for which you want to display information from the list of files in the client area.
2. Select `Details` from the Show menu. The CMVC - Information window is opened and the file details are displayed.

Note: If you do not want to select files from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Showing Differences in Files on CMVC

You can specify two files that exist in CMVC, or two different versions of the same file, and view the differences between their content.

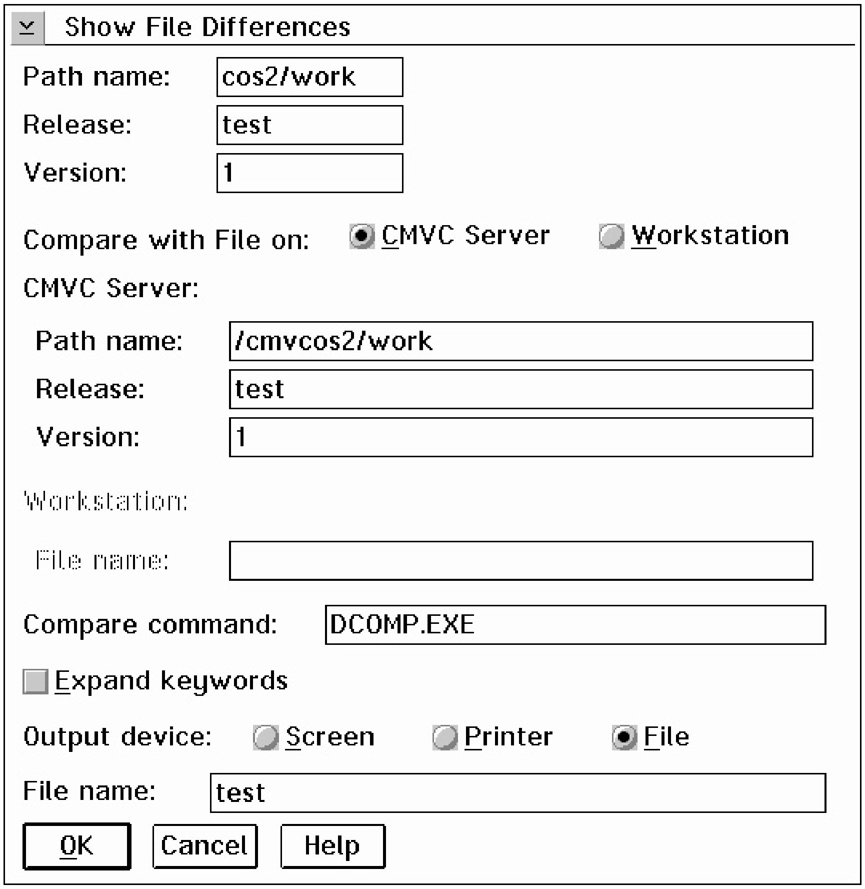
Authority Required: The component owner has implicit authority. You have explicit authority if the FileExtract action is specified for your user ID in the component associated with the file.

Prerequisites: When showing file differences, you must ensure that the following requirements are met:

- Versions of the specified file must exist.
- If you use the default compare command (**dcomp.exe**), the file must be a text file.
- The file is not marked as deleted.

Procedure: To show the differences between files:

1. Select the file you want to compare to from the list of files in the client area.
2. Select `Differences...` from the Show menu to display the Show File Differences dialog box (see Figure 7).



Show File Differences

Path name: cos2/work

Release: test

Version: 1

Compare with File on: ● CMVC Server ○ Workstation

CMVC Server:

Path name: /cmvcos2/work

Release: test

Version: 1

Workstation:

File name:

Compare command: DCOMP.EXE

□ Expand keywords

Output device: ○ Screen ○ Printer ● File

File name: test

Figure 7. Show File Differences Dialog Box

Accept the prefilled information in the `Path name` and `Release` fields or type new information.

If the files that you want to compare have different path names, change the path name specified in the `Path names` field under the CMVC Server section.

If you want to compare common files, enter the name of the release for the common file in the `Release` field under the CMVC Server section.

3. Indicate the versions of the files that you want to compare in the `Version` fields. The first and second `Version` fields are optional. You do not have to enter the version for the second file. This allows you to compare the current version of a file belonging to two releases. If the files are identical, a message is displayed.

4. Specify the compare command to be performed in the `Compare command` field. The maximum length of the compare command is 195 characters.

5. If you want to substitute assigned values in place of keywords, select `Expand keywords`. The default is not to substitute values for the keywords.

6. Specify the output device on which you want the differences shown. The output devices include:

   Screen Displays the results of the comparison in the CMVC - Information window. This is the default device.

   Printer Prints the results of the comparison.

File Stores the results of the comparison in a file.

When you select this button, the `File name` field at the bottom of the dialog box is activated. If you want to create a file on your workstation, type a file name. The name can consist of directory names and a base file name. You cannot use blanks, tabs, or control characters. The maximum length of the output file name is 1023 characters.

Note: The **Printer** and **File** options are available only if you use **DCOMP.EXE**. as your compare command.

7. Select `OK` to show the file differences.

Note: If you do not want to select files from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Showing Differences in Client Workstation and Server Files

You can compare a file that is in CMVC with a file that exists in a directory on your client workstation and display the differences.

Authority Required: The component owner has implicit authority. You have explicit authority if the FileExtract action is specified for your user ID in the component associated with the file.

Prerequisites: When you show differences between client workstation and server files, you must ensure that the following requirements are met:

- Version of the file is on CMVC.
- File is not marked as deleted.
- You have read permission for the file that is on your workstation and for the directory in which the file is stored.
- If you use the default compare command (**dcomp.exe**), the file must be a text file.

Procedure: To show differences between client workstation and server files:

1. Select the file that you want to compare with another file from the list of files in the client area.
2. Select `Differences...` from the Show menu to display the Show File Differences dialog box. Accept the prefilled information in the `Path names` and `Release` fields or type new information.
3. If you want to compare files that are on the client workstation, select `Workstation`. The default is to compare files that are located on the CMVC server.

   The `File name` field in the `Workstation` section is activated, and the fields under the `CMVC Server` section are deactivated. The `File name` field is prefilled with the values in the `Directory` option and the `Path names` field.

   The file name is the name of the file on your client workstation. This can consist of directory names and a base file name. You cannot use blanks, tabs, or control characters. The maximum length of the file name is 1023 characters.
4. Specify the version of the file on CMVC that you want to compare to the client workstation file. By default, the current version of the file is compared.

5. Specify the compare command to be performed. The maximum length of the compare command is 195 characters.
6. If you want to substitute assigned values in place of keywords, select `Expand keywords`. The default is not to substitute values for the keywords.
7. Specify the output device on which you want to display the differences.

   Screen Displays the results of the comparison in the CMVC - Information window. This is the default device.

   Printer Prints the results of the comparison.

   File Stores the results of the comparison in a file.

   When you select this button, the `File name` field at the bottom of the dialog box is activated. If you want to create a file on your client workstation, type a file name in the entry field. The maximum length of the output file name is 1023 characters. The name can consist of directory names and a base file name. You cannot use blanks, tabs, or control characters.
8. Select `OK` to display the differences.

Note: If you do not want to select files from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Showing the Change History of a File

You can show the history of file changes.

Authority Required: None.

Prerequisites: The file exists in CMVC.

Procedure: To show the change history of a file:

1. Select the file for which you want to display the change history from the list of files in the client area.
2. Select `Change history` from the Show menu. The CMVC - File Change History window is opened and the information that you requested is displayed.

Chapter 4. Working with Defects

You can use the CMVC - Defects window to perform a variety of actions on defects. This chapter describes the actions that you can perform as a defect *originator* and *owner*. If you have superuser authority, you can perform any of the actions listed in this chapter.

For more information about the role defects play in organizing and controlling changes, refer to the book *IBM CMVC Concepts*. The actions that a *sizing record* owner performs in relation to a defect and that a *verification record* owner performs in relation to a defect are described in Chapter 6, “Working with Sizing Records and Verification Records” on page 83. The actions that a track owner performs are described in Chapter 7, “Working with Tracks” on page 97, and the actions that a level owner performs are discussed in Chapter 8, “Working with Levels and Level Members” on page 111.

Defect Originator’s Tasks

Defects are opened to report a problem in a project under CMVC control. When you open a defect, you become the defect originator by default, and can therefore perform certain additional actions.

Some of the actions that you can perform as a defect originator include:

- Open, cancel, or reopen a defect
- Verify a defect
- Add remarks to a defect
- Modify the originator or the properties of a defect
- Reassign a defect to another component
- Show the details of a defect
- Show the process configurations of a defect
- Show the change history of a defect.

States of a Defect

A defect monitors reported problems. The life cycle of a defect has 9 possible states:

- Open
- Design
- Size
- Review
- Working
- Verify
- Returned
- Canceled
- Closed.

Note: If a CMVC subprocess, such as tracking, is not included in the process for the release, the state associated with the subprocess is bypassed, provided that there are no dependencies or that you do not have to force the object to the next state. For more information about CMVC subprocesses, refer to the book *IBM CMVC Server Administration and Installation*.

The life cycle of the defect begins when you open the defect (`open` state). As the originator, you can cancel a defect if it is in the `open` or `returned` state. You may decide to cancel a defect if it refers to a problem that has been addressed.

If you specified a release when you opened the defect, the defect moves from the `working` state to the `verify` state when the track for that release reaches the `complete` state.

If you did not specify a release, the defect moves from the `working` state to the `verify` state when one of the tracks for that defect reaches the `complete` state.

The defect moves from the `verify` to the `closed` state when all verification records associated with this defect are marked and all tracks for the defect reach the `complete` state.

Displaying Defect Information in the Client Area

The CMVC - Defects window allows you to create and work with defects. When you query a defect, specific information is displayed in the client area of the window under the following headings:

Prefix Displays the prefix specified when the defect was opened. Your family administrator can configure a list of prefixes for your development environment.

Number Displays the identifier of the defect.

Component Displays the name of the component that manages the defect.

State Displays the current state of the defect.

Originator Displays the user ID of the defect originator.

Owner Displays the user ID of the defect owner.

Severity Indicates the severity of the problem to be resolved. Your family administrator can configure these values.

Age Displays the time since the defect was opened.

Reference Displays an optionally assigned value that groups related defects and features.

Abstract Displays a summary of the problem that addresses the defect that you opened.

Priority Indicates the timing or scheduling requirements for closing a defect.

Target Indicates when the work on a defect will be completed (a level or a date, for example).

Note: Other fields may also be displayed if your family administrator has configured additional fields. Certain shipped fields may also be suppressed or renamed.

For more information about configured fields, refer to “Working with Configurable Fields” on page 22, the book *IBM CMVC Server Administration and Installation*, or see your family administrator.

Performing Queries with the Open Defect List

The Open Defect List dialog box allows you to perform queries and list information in the CMVC - Defects window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to “Defining Default Queries and Tasks” on page 19.

To perform a query using the Open Defect List dialog box:

1. Select `Open list...` from the File menu to display the Open Defect List dialog box.
2. Enter the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

   Prefixes Prefixes specified when the defect was opened. Your family administrator can configure a list of choices for your development environment. Use the `Choices...` button to make your selection from a list of available alternatives.

   Defects Identifier of the defect. You can type up to 15 alphanumeric characters.

   Components Component that manages the defect. For example, you may want to list information about the defects that are referenced by the `tools` component.

   States Particular state or states of a defect. Valid states are open, design, size, review, working, verify, returned, canceled, and closed.

   Originator IDs User ID of the defect originator. For example, you may to list the defects that `mikes` originated.

   Owner IDs User ID of the defect owner. For example, you may want to list the defects that `patjones` owns.

   Severities Value that indicates the severity of the problem that the defect addresses. Your family administrator can configure a list of choices. Use the `Choices...` button to make your selection from a list of available alternatives.

   Ages Time that has elapsed since the defect was opened. Your family administrator can determine the algorithm for computing this age.

   References Optionally assigned value for a defect or feature that is used to group related defects or features.

   Abstracts Summary of the problem that the defect addresses. For example, you may want to list all defects which have the keyword `resource` in them by using the SQL wildcard character (`%`) along with the `like` operator, and search for the string `%resource%`.

Releases Name of the releases that are affected by the defects. For example, you may want to search for defects that affect release `debugger4`.

Duplicates Defect that is specified by more than one defect or feature. For example, you may want to search for defects that have the defect `APAR1159` specified as a duplicate.

Add dates Date (yy/mm/dd) on which the defect was created. For example, you may want to list only the defects created after `93/04/29`. Valid SQL operators are `<`, `>`, `like`, `is null`, and `between`.

Answers Answers that you provide when accepting a defect. Your family administrator can add new values to this field. Use the `Choices...` button to make your selection from a list of available alternatives.

Levels Name of the levels in which the problem was discovered.

Environments Name of the environments in which the problem was discovered. You can type up to 15 alphanumeric characters. You can use all characters except vertical bars (|), ASCII control characters or shell metacharacters.

Symptom Value that indicates the symptom of the defect. Your family administrator can suppress this field or add new values.

Phase found Value that indicates the development phase that was in progress when the defect was discovered. Your family administrator can suppress this field or add new values.

Phase injected

Value that indicates the development phase that was in progress when the defect was injected in the code. Your family administrator can suppress this field or add new values.

Priority Value that indicates the urgency for resolving the defect. Your family administrator can suppress this field or add new values. Use the `Choices...` button to make your selection from a list of available alternatives.

Target Reference, such as a date, for the work being done on the defect. For example, you may want to see the defects for which you expect work to be completed by `93/04/22`. Your family administrator can suppress this field or add new values.

3. Type the required information in any additional entry fields that have been configured by your family administrator. If your family administrator has suppressed certain shipped fields, some of the fields described above may not be displayed.

4. Select `OK` to perform the query. The CMVC - Defects window is refreshed, and the information that you requested is displayed in the client area.

You can use the `Refresh now` item on the File menu to reissue a query to the server. The most recent query is sent and the list is updated with current information from the server. If you issue a query to the server using the `Open list...` option, you can replace the query with the default query specified in the system file by selecting the `Default list` item from the File menu.

Note: For more information about how to use the Options menu items to specify operators, refer to “SQL Operator (in) and No Sort Combo Boxes” on page 11.

Performing Originator Actions with the CMVC - Defects Window

This section describes the authority and prerequisites for performing common actions with the CMVC - Defects window, and how to perform these actions.

Opening a Defect

When you find a problem in a project under CMVC control, you can report it by opening a defect to ensure that other people in your group are aware of it.

Authority Required: None.

Prerequisites: When opening a defect, you must ensure that the following requirements are met:

- You have a valid CMVC user ID for the CMVC family in which a problem exists.
- You must know the name of the component affected by the problem.

Procedure: To open a defect:

1. Select `Open...` from the Actions menu to display the Open Defect dialog box.
2. Type information in the following fields:

   Component Component that the defect addresses. By default, the owner of this component becomes the owner of the defect. You can type `root` and allow the root component owner to assign the defect to the appropriate component. You can also import a selected component from the CMVC - Components window.

   Remarks Detailed description of the problem that the defect addresses. An example is

   `The widget is not producing the desired output.`

   You can type up to 15 999 alphanumeric characters. Select `Edit...` to open the Remarks Editor.

   Abstract Summary of the problem that the defect addresses. An example is `Non-standard characters on display`. If you do not enter anything in this field, the default is to use either the first 63 characters or the first line of information from the `Remarks` field. Typing in an abstract could be more meaningful than relying on the default. Do not use a vertical bar (|), `$ ",`` ` ``, or \, as these have a special meaning to the shell.

   Number Identifier of the defect. You can type up to 15 alphanumeric characters. If you leave this field empty, CMVC assigns a number to the defect and informs you of the number.

   Release Name of an existing release to trigger the defect verification process. You can also import a selected release from the CMVC - Releases window. When the tracks for this release move to the `complete` state, then the defect verification process begins. If this field is left empty, the verification process will begin when the first track for the defect is complete.

Level Name of an existing level in which the problem was discovered. You can import a selected level from the CMVC - Levels window.

Environment Name of the environment in which the problem was discovered. Examples are `modelA` or `PCversion2`. You can type up to 15 alphanumeric characters. You can use all characters except vertical bars (|), ASCII control characters or shell metacharacters. You can also import a selected environment list entry from the CMVC - Environment Lists window.

Reference Optionally assigned value that groups related defects and features. You can type up to 15 alphanumeric characters. You can use all characters except vertical bars (|), ASCII control characters or shell metacharacters.

Prefix Prefix that indicates the type of defect. Your family administrator can suppress this field or add new values. Use the `Choices...` button to select from a list of available alternatives.

Severity Value that indicates the severity level of the defect. Your family administrator can suppress this field or add new values. Use the `Choices...` button to select from a list of available alternatives.

Symptom Value that indicates the symptom of the defect. Your family administrator can suppress this field or add new values.

Phase found Value that indicates the development phase that was in progress when the defect was discovered. Your family administrator can suppress this field or add new values.

3. Type the required information in any additional entry fields that have been configured by your family administrator. If your family administrator has suppressed certain shipped fields, some of the fields described above may not be displayed.

   Note: For some of these configurable fields, you can also use the `Choices...` button to make your selection from lists of alternatives.

4. Select `OK` to open the defect. An information box displays a message similar to the following:

   ```
   A new defect was opened successfully.
   The new defect number is 1159.
   ```

The value displayed is a unique identifier for the defect you just opened. You can refresh the list in the CMVC - Defects window to see the new defect and perform further actions on it.

The defect is in the `open` state and you become the originator of the defect. The owner of the component against which you opened the defect becomes the owner of the defect, and must perform one of the following actions:

- Move the defect to the `design` state if the defect design, size, and review (DSR) subprocess is specified for the associated component process
- Accept the defect if the defect design, size, and review (DSR) subprocess is not specified for the associated component process

- Return it to the originator
- Assign it to a different component or user.

Reopening a Defect

You can reopen a defect for the following reasons:

- The defect was returned. In this case you reopen it so that you can give a more detailed explanation of the problem and the need to resolve it.
- You canceled a defect, but now you want it resolved.

Authority Required: The defect originator has implicit authority. You have explicit authority if the DefectReopen action is specified for your user ID in the component associated with the defect.

Prerequisites: When reopening defects, you must ensure that the defect is in the `canceled` or `returned` state.

Procedure: To reopen a defect:

1. Select the defect you want to reopen from the list of defects in the client area.
2. Select `Reopen...` from the Actions menu to display the Reopen Defects dialog box. Accept the prefilled information in the `Defects` field or type new information.
3. Type any additional comments, such as the reason that you are reopening the defects, in the `Remarks` field. You can type up to 15 999 alphanumeric characters. An example is :

   ```
   It wasn't a user error after all.
   New information indicates a widget problem.
   ```

4. Select `OK` to reopen the defect. The defect moves to the `open` state. From this state, the defect owner can accept or return it to the originator.

Note: If you do not want to select defects from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Canceling a Defect

If a defect is not valid or it is a duplicate of an existing defect or feature, the defect originator can move it to the `canceled` state.

Authority Required: The defect originator has implicit authority. You have explicit authority if the DefectCancel action is specified for your user ID in the component associated with the defect.

Prerequisites: The defect is in the `open` or `returned` state.

Procedure: To cancel a defect:

1. Select the defect that you want to cancel from the list of defects in the client area.
2. Select `Cancel...` from the Actions menu to display the Cancel Defects dialog box. Accept the prefilled information in the `Defects` field or type new information.

3. Type additional comments in the Remarks field. You can type up to 15 999 alphanumeric characters. An example is I opened this defect prematurely, the problem was a user error.
4. Select OK to cancel the defect. The defect moves to the canceled state. This is the final state of a defect that is rejected or a duplicate defect, therefore, no further action is required. The originator, however, can reopen a defect that is in the canceled state, if necessary.

Note: If you do not want to select defects from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Verifying a Defect

If no tracks exist, the defect owner must move the defect to the verify state when work on it is completed. If there are tracks, the defect moves from the working to the verify state automatically.

If a defect's Release field is empty, it changes from the working to verify state once the first track associated with the defect is complete. If a defect's Release field is prefilled with a release name, then it changes from the working to verify state when the track associated with the release is complete.

Authority Required: The defect owner has implicit authority. You have explicit authority if the DefectVerify action is specified in your user ID in the component associated with the defect.

Prerequisites: When you verify a defect, you must ensure that the defect is in the working state.

Procedure: To move defects to the verify state:

1. Select the defects you want to force to the verify state from the list of defects in the client area.
2. Select Verify... from the Actions menu to display the Verify Defects dialog box. Accept the prefilled information in the Defects field or type new information.
3. Enter additional comments in the Remarks field. You can type up to 15 999 alphanumeric characters. An example is:

   ```
   I forced this defect from working state to verify state
   because there are no tracks associated with this defect
   and I want the originator to verify the results using a
   verification record.
   ```

4. Select OK to move the defect to the verify state.

The following occurs:

- The defect moves to the verify state.
- Any existing verification records move from the notReady state to the ready state. (There may be more than one verification record if duplicate defects exist. The verification records of those returned duplicates are activated by the movement of the one active defect.)
- Owners of verification records are notified and they must mark their verification records before the defect can move to the closed state. For more information,

refer to Chapter 6, “Working with Sizing Records and Verification Records” on page 83.

Note: If you do not want to select defects from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Adding Remarks to a Defect

You can add remarks or comments to a defect.

Authority Required: None.

Prerequisites: The defect exists in any state.

Procedure: To add remarks to a defect:

1. Select the defect to which you want to add remarks from the list of defects in the client area of the CMVC - Defects window.
2. Select `Add remarks...` from the Actions menu to display the Add Defect Remarks dialog box. Accept the prefilled information in the `Defects` field or type new information.
3. Type your comments in the `Remarks` field. An example is:

   ```
   I heard that Joe Hamilton’s development team
   experienced a similar widget problem.
   ```

   If you require additional space for your remarks, select `Edit...` to display the Remarks Editor dialog box. You can type up to 15 999 alphanumeric characters.
4. Select `OK` to save the remarks to the server. The remarks cannot be edited or removed.

Note: If you do not want to select defects from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Reassigning a Defect to Another Component

If a defect is opened against a component but the defect actually affects another component, you can reassign the defect to the other component.

Authority Required: The defect originator and owner have implicit authority. You have explicit authority if the DefectAssign action is specified for your user ID in the component associated with the defect.

Prerequisites: If you are the defect originator, you can only reassign a defect that is in the open state.

Procedure: To reassign a defect:

1. Select the defect for which you want to specify a new component from the list of defects in the client area.
2. Select `Component...` from the Modify menu to display the Modify Defect Component dialog box. Accept the prefilled information in the `Defects` field or type new information.
3. Enter information in the following fields:

New component Name of the component to which you want to assign the defects. You can import a selected component from the CMVC - Components window.

New Owner User ID for the new defect owner. You can import a selected user ID from the CMVC - Users window.

Remarks Additional comments about the modification. You can type up to 15 999 alphanumeric characters. For example:

```
This defect does not pertain to the component that it
was opened to address. I am reassigning defect 1159
to the component that manages widget development.
```

4. Select `OK` to reassign the defect. The defect is associated with another component. The owner of that component or the person you specify becomes the new defect owner and can perform certain tasks.

Note: If you do not want to select defects from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Modifying the Originator of a Defect

You can assign the role of defect originator to another CMVC user ID.

Authority Required: The defect originator has implicit authority. You have explicit authority if the DefectModify action is specified for your user ID in the component associated with the defect.

Prerequisites: When modifying the defect originator, you must ensure that the new originator has a valid CMVC user ID. The defect can be in any state.

Procedure: To modify the defect originator:

1. Select the defects for which you want to specify a new originator from the list of defects in the client area.
2. Select `Originator...` from the Modify menu to display the Modify Defect Originator dialog box. Accept the prefilled information in the `Defects` field or type new information.
3. Type the user ID of the person you want to be the new originator of the specified defects in the New originator field. You can import a selected user ID from the CMVC - Users window.
4. Type additional comments about the modification in the `Remarks` field. You can type up to 15 999 alphanumeric characters. An example is :

   ```
   I am moving to another department and am relinquishing
   responsibility of defect 1159. I've filled Mary Peters
   in on her new responsibilities.
   ```

5. Select `OK` to modify the defect originator. The new originator assumes responsibility for verifying the outcome of the defect.

Specifying a new originator does not automatically modify the verification record created for the originator. You must reassign the originator's verification record manually, as required.

Note: If you do not want to select defects from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Modifying the Properties of a Defect

The defect originator or owner can modify certain properties of a defect, such as, the abstract, the number, or the reference.

Authority Required: The defect originator and owner have implicit authority. You have explicit authority if the DefectModify action is specified for your user ID in the component associated with the defect.

The defect originator also has implicit authority to modify the severity rating of a defect. You have explicit authority to modify the priority of a defect if the DefectModify action has been specified for your user ID in the component associated with the defect.

Prerequisites: The defect can be in any state.

Procedure: To modify the properties of a defect:

1. Select the defects for which you want to modify certain properties from the list of defects in the client area.
2. Select `Properties...` on the Modify menu to display the Modify Defect Properties dialog box. Accept the prefilled information in the `Defects` field or type new information.
3. Type information in the following fields:

   Abstract A restatement of the summary of the problem the defect was opened to address. The abstract must not exceed 63 characters. An example is `Incorrect output`. Do not use a vertical bar (|), $, ", `, or \, because these have a special meaning to the shell.

   Number New number for your defect. You can type up to 15 alphanumeric characters.

   Release Name of an existing release to trigger the defect verification process. You can import a selected release from the CMVC - Releases window. When the tracks for this release move to the `complete` state, defect verification can begin.

   Level Name of an existing level in which the problem was discovered. You can import a selected level from the CMVC - Levels window.

   Environment Name of the environment in which the problem was discovered. You can type up to 15 alphanumeric characters. You can use all other characters except vertical bars (|), ASCII control characters or shell metacharacters.

   Reference Optionally assigned value that groups related defects and features. You can type up to 15 alphanumeric characters. You can use all other characters except vertical bars (|), ASCII control characters or shell metacharacters.

   Prefix A value that indicates the type of defect. Your family administrator can configure these values. You can also use the `Choices...` button to make your selection from a list of available alternatives.

Severity A value that indicates the severity of the defect. Your family administrator can configure these values. You can also use the `Choices...` button to make your selection from a list of available alternatives.

Answer Answer that you provide when accepting a defect. Your family administrator can add new values to this field. You can use the `Choices...` button to select from a list of available alternatives.

Remarks Additional comments about the defect. You can type up to 15 999 alphanumeric characters. An example is:

```
This defect requires some code drops in files debug.48
and debug2.3.
```

Symptom Value that indicates the symptom of the defect. Your family administrator can suppress this field or add new values.

Phase found Value that indicates the development phase that was in progress when the defect was discovered. Your family administrator can suppress this field or add new values. Use the `Choices...` button to select from a list of available alternatives.

Phase injected Development phase in progress when the defect was injected in the code. Your family administrator can suppress this field or add new values.

Priority Value that indicates the urgency for resolving the defect. Your family administrator can suppress this field or add new values.

Target Value that indicates a target date for resolving the defect. Your family administrator can suppress this field or add new values.

4. Type the required information in any additional entry fields that have been configured by your family administrator. If your family administrator has suppressed certain shipped fields, some of the fields described above may not be displayed.

5. You can also add comments in the `Remarks` field to indicate the reason for changing one or more properties. You can type up to 15 999 alphanumeric characters. An example is:

```
The symptoms of defect 1159 indicate a more severe
problem than first expected.  I've changed the
severity to a value of 1.
```

6. Select `OK` to modify the properties. Any modifications made are reflected in the updated defect.

Note: If you do not want to select defects from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Showing the Details of a Defect

You can show details about a defect, including the:

- Component to which the defect belongs
- Current state of the defect
- Add and change dates
- Names of the originator and the owner
- Tracks addressing the defect, if any
- Status of any existing verification records
- History of the actions made and remarks added to the defect
- Process configurations of a defect.

Authority Required: The defect originator has implicit authority. You have explicit authority if the DefectView action is specified for your user ID in the component associated with the defect.

Prerequisites: The defect can be in any state.

Procedure: To show details for a single defect, double-click on a defect listed in the client area.

To show details for more than one defect:

1. Select the defects from the list of defects in the client area.
2. Select `Details` from the Show menu. The CMVC - Information window is opened and the details are displayed.

Note: If you do not want to select defects from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Showing the Process Configurations of a Defect

You can show the configuration of the processes that are associated with the defect.

Authority Required: The defect owner has implicit authority. You have explicit authority if the DefectView action is specified for your user ID in the component associated with the defect.

Prerequisites: The defect can be in any state.

Procedure: To show the process configurations for a defect:

1. Select the defect for which you want to show the process configurations from the list of defects in the client area.
2. Select `Process configurations` from the Show menu. The CMVC - Information window is opened and the configured processes are displayed.

Note: If you do not want to select defects from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Showing the Change History of a Defect

You can show the history of changes for defects. For more information about how to show the change history, refer to “Using the CMVC - Change History Window” on page 23.

Additional Show Options

Additional show options allow you to open certain GUI windows, including the CMVC - Sizing Records, CMVC - Verification Records, CMVC - Tracks, CMVC - Approval Records, CMVC - Fix Records, CMVC - Test Records and CMVC - Level Members windows. To open these windows, select the corresponding option from the show menu.

Defect Owner’s Tasks

When a user in your CMVC family opens a defect against a component you own, or assigns an existing defect to your CMVC user ID, you become the owner of the defect. You can either retain ownership of the defect or reassign ownership to another user ID in your family.

Some of the actions that you can perform as a defect owner include:

- Design, review, or size a defect
- Accept or return a defect
- Reassign a defect to another component
- Modify the ownership or properties of a defect
- Show the details of a defect
- Create a track
- Show the process configurations of a defect.

For a description of how to modify additional properties of a defect and how to show the process configurations, refer to “Defect Originator’s Tasks” on page 45.

Designing a Defect

You can move a defect to the `design` state to add specific design information.

Authority Required: The defect owner has implicit authority. You have explicit authority if the DefectDesign action is specified for your user ID in the component associated with the defect.

Prerequisites: The defect can be in either the `open`, `design`, `size`, `review`, or `returned` state.

Procedure: To design a defect:

1. Select the defects you want to move to the `design` state from the list of defects in the client area.
2. Select `Design...` from the Actions menu to display the Design Defects dialog box. Accept the prefilled information in the `Defects` field or type new information.
3. Type the design information, such as, the defect resolution text, in the `Remarks` field. You can type up to 15 999 alphanumeric characters. An example is `Reorder the fields in the Compose Music dialog box.`

4. Select OK to move the defect to the design state. This occurs if the defect was in any of the open, returned, size, or review states.
5. If you moved a defect from the review state to the design state, you must re-mark any sizing records.

You can move a defect from the design state to:

- The design state so that you can enter additional information, as often as required.
- The size state to indicate which resources are required to resolve the defect.
- The returned state if you believe that the defect should not or can no longer be resolved.

Note: If you do not want to select defects from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Sizing a Defect

You can move a defect to the size state so that the effort to resolve the defect can be sized.

Authority Required: The defect owner has implicit authority. You have explicit authority if the DefectSize action is specified for your user ID in the component associated with the defect.

Prerequisites: When sizing a defect, you must ensure that the defect is in the design state and that design details are entered as text.

Procedure: To size a defect:

1. Select the defects you want to size from the list of defects in the client area.
2. Select Size... from the Actions menu to display the Size Defects dialog box. Accept the prefilled information in the Defects field or type new information.
3. Type additional comments, such as sizing text, in the Remarks field. You can type up to 15 999 alphanumeric characters. An example is Joan, ask Peter to help you with the estimates.
4. Select OK to move the defect to the size state. Any sizing records that exist for that defect will move to the ready state. You can now create additional sizing records. For more information about creating sizing records, refer to “Creating a Sizing Record” on page 85.

You can move a defect from the size state to:

- The review state
- The design state
- The returned state.

Note: If you do not want to select defects from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Reviewing a Defect

You can move a defect to the `review` state from the `size` state so that a formal review of the proposed defect resolution and its sizing information can take place.

Authority Required: The defect owner has implicit authority. You have explicit authority if the DefectReview action is specified for your user ID in the component associated with the defect.

Prerequisites: All sizing records are marked.

Procedure: To move a defect to the `review` state:

1. Select the defects for which you want the resolution and sizing information formally reviewed from the list of defects in the client area.
2. Select `Review...` from the Actions menu to display the Review Defects dialog box. Accept the prefilled information in the `Defects` field or type new information.
3. Type additional comments about the defects that you want reviewed in the `Remarks` field. You can type up to 15 999 alphanumeric characters. An example is:

   ```
   Keep in mind that two programmers are on holidays
   next month.
   ```

4. Select `OK` to move the defect to the `review` state.

You can return the defect from the `review` state to the originator if you do not want to resolve it. You can accept the defect for resolution and thereby change the state of the defect to `working`. You can move a defect from the `review` state to the `design` state if additional design information is required. If you do so, you will have to remark the sizing records.

Note: If you do not want to select defects from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Accepting a Defect for Resolution

You can accept a defect as being valid and move the defect to the `working` state.

Authority Required: The defect owner has implicit authority. You have explicit authority if the DefectAccept action is specified for your user ID in the component associated with the defect.

Prerequisites: When accepting a defect for resolution, you must ensure that:

- If the design, size, and review subprocess is included in the process for the component, the defect is in the `review` state.
- If the design, size, and review subprocess is not included in the process for the component, the defect is in the `open` or `returned` state.

Procedure: When accepting a defect:

1. Select the defects you want to accept from the list of defects in the client area.
2. Select `Accept...` from the Actions menu to display the Accept Defects dialog box. Accept the prefilled information in the `Defects` field or type new information.

3. Type information in the following text entry fields:

   Answer Answer to the defects. Examples are comply_with or remove_code. Your family administrator can configure a list of choices. You can also use the Choices... button to make your selection from a list of available alternatives.

   Remarks Additional comments about the defect. You can type up to 15 999 alphanumeric characters. An example is:

   ```
   This defect requires some code drops in files debug.48
   and debug2.3.
   ```

4. Select OK to accept the defect. The defect moves to the working state.

At this time, one track will be created automatically for every release that has a sizing record in the accept state. Additional tracks can be created at this time for releases for which the tracking subprocess is specified. (See "Creating a Track" on page 62.) A verification record is created in the notReady state for the defect originator.

Note: If you do not want to select defects from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Returning a Defect

You can return a defect to the originator if the defect is not valid or a solution is not feasible. You can also return a defect if another defect or feature already exists that addresses the same issue.

Authority Required: The defect owner has implicit authority. You have explicit authority if the DefectReturn action is specified for your user ID in the component associated with the defect.

Prerequisites: When returning a defect, you must ensure that the following requirements are met:

- The defect is in the open, design, size, or review states.
- If the defect is in the working state, the defect must have no tracks associated with it.

Procedure: To return a defect:

1. Select the defect that you want to return from the list of defects in the client area.
2. Select Return... from the Actions menu to display the Return Defects dialog box. Accept the prefilled information in the Defects field or type new information.
3. Select one of the Duplicates or Answer radio buttons to indicate whether you are returning the defects as duplicates or with an answer. By default, the Answer button is selected.
4. If you use the default, you must type the reason for returning the defects in the Answer entry field. Your family administrator can configure these values. Examples are future or as_designed. Use the Choices... button to make your selection from a list of available alternatives.
5. If you are returning the defects as duplicates, you must type the duplicate defect or feature identifier in the Duplicate entry field. This identifier

addresses the same problem as the defects that you are returning. That defect or feature must not be currently in the `canceled`, `closed`, or `returned` state.

6. Type additional comments about the defects that you are returning. You can type up to 15 999 alphanumeric characters. An example is:

   ```
   This defect cannot be reproduced. Please
   provide more details about your working procedure.
   ```

7. Select `OK` to move the defect to the `returned` state.

The originator must address the defect when it is in the `returned` state. Possible actions include:

- Reopening it to the same component, with additional information
- Reopening it to another component that is affected by the defect
- Canceling it if it does not need to be resolved.

However, if you change your mind, you can accept a defect while it is in the `returned` state, if the defect design, size, and review (DSR) subprocess is not included in the process for the component.

Note: If you do not want to select defects from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Creating a Track

Tracks are required for defects for which the *tracking subprocess* is included in the process for the release. A track is created automatically in defects for each sizing record that is in the `accept` state. Otherwise, a track has to be created manually. The track life cycle begins when the defect is in the `working` state.

A track does not have a single name. Rather, a track is referred to by specifying a release name and a defect name.

Note: You can also use the CMVC - Tracks window to create a track.

Authority Required: The defect owner has implicit authority. You have explicit authority if the TrackCreate action is specified for your user ID in the component associated with the defect.

Prerequisites: When creating tracks, you must ensure that the following requirements are met:

- The tracking subprocess is included in the process for the release
- The defect is in the `working` or `verify` state.

Creating tracks is a mandatory task for defects if the process of the associated release includes the tracking subprocess. Creating a track leads to more effective change control.

Procedure: To create tracks:

1. Select the defect for which you want to create a track from the list of defects in the client area.

2. Select `Create Tracks...` from the Actions menu of the CMVC - Defects window to display the Create Tracks dialog box. Accept the prefilled information in the `Defects` field or type new information.

3. Type information in the following fields:

Releases Name of existing releases in which changes will be tracked to resolve the defects. You can also import selected releases from the CMVC - Releases window.

Target A reference, such as a level or a date, for the work being done for the tracks. You can use all characters except vertical bars (|), ASCII control characters or shell metacharacters. You can type up to 15 alphanumeric characters.

Owner User ID of the track owner. If you do not specify an owner, then the default is the user creating the track. You can import a selected user ID from the CMVC - Users window.

4. Select `OK` to create the track.

One track is created for every specified defect in every specified release. For example, four tracks are created if you specify defects `155` and `1323`, and releases `code1` and `books1` in the Create Tracks dialog box. The following tracks are included:

- Two tracks that refer to fixes required in the release for source code
- Two tracks that refer to the same fixes required in the release for books documenting the application generated by that source code.

If there are any sizing records in the `accept` state when a track is created automatically, then one fix record is created for each of those sizing records. No fix records are created for sizing records in the `reject` or `abstain` states.

Because you created the track, you become the owner of the track associated with that release, unless a different user is specified. As the track owner, you can perform certain tasks. For more information on the tasks for which a track owner is responsible, refer to Chapter 7, “Working with Tracks” on page 97.

While the defect is in the `working` or `verify` state, you can create additional tracks if you find that the same defect applies to other releases. For example, defect 1159 was opened to correct a problem that applies to release4. When you examine the problem in more detail, you discover that older maintenance releases (`release2` and `release3`) also need this fix applied to them.

Note: As a defect owner who created a track, you can cancel that track. If you reassign a track to another owner, the new track owner does not have implicit authority to cancel the track. For more information about canceling a track refer to “Canceling a Track” on page 102.

Modifying the Owner of a Defect

You can assign a defect to another CMVC user ID, thereby transferring the responsibility of defect ownership.

Authority Required: The defect owner and originator have implicit authority. You have explicit authority if the DefectAssign action is specified for your user ID in the component associated with the defect.

Prerequisites: When modifying defect ownership, you must ensure that the following requirements are met:

- New owner must have a valid CMVC user ID.
- Defect exists in any state.

- If you are the defect originator, you can only reassign the defect to a new owner if it is in the open state.

Procedure: To modify defect ownership:

1. Select the defects for which you want to modify the owner from the list of defects in the client area.
2. Select Owner... from the Modify menu to display the Modify Defect Owner dialog box. Accept the prefilled information in the Defects field or type new information.
3. Type information in the following fields:

   New owner User ID of the new owner of the specified defects. You can also import a selected user ID from the CMVC - Users window.

   Remarks Additional information relating to the defect. You can type up to 15 999 alphanumeric characters. An example is:

   ```
   Chris Stevens will assume ownership of defect
   1159 because I am going on an extended leave of
   absence to cycle from Toronto to Kapuskasing.
   ```
4. Select OK to modify the owner. The new owner assumes the responsibilities of the defect owner.

Note: If you do not want to select defects from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Chapter 5. Working with Features

You can use the CMVC - Features window to perform a variety of actions on features. This chapter describes the actions that you can perform as a feature originator and feature owner. If you have superuser authority, you can perform any of the actions listed in this chapter.

For more information about the role features play in implementing changes, refer to the book *IBM CMVC Concepts*.

The actions that a sizing record owner performs in relation to a feature and that a verification record owner performs in relation to a feature are described in Chapter 6, “Working with Sizing Records and Verification Records” on page 83. The actions that a track owner performs are described in Chapter 7, “Working with Tracks” on page 97 and the actions that a level owner performs are discussed in Chapter 8, “Working with Levels and Level Members” on page 111.

Feature Originator’s Tasks

Features are opened to suggest a design change or an enhancement to a project under CMVC control. When you open a feature you become the originator of the feature by default, and can therefore perform certain additional actions.

Some of the actions that you can perform as a feature originator include:

- Open, cancel, or reopen a feature
- Verify a feature
- Add remarks to a feature
- Modify the originator or the properties of a feature
- Show the details of a feature
- Show the process configurations of a feature
- Show the change history of a feature.

States of a Feature

A feature monitors proposed changes in the same way as a defect monitors reported problems. Similarly, the life cycle of a feature has 9 possible states:

- Open
- Design
- Size
- Review
- Working
- Verify
- Returned
- Canceled
- Closed.

Note: If a CMVC subprocess, such as tracking, is not included in the process for the release, the state associated with the subprocess is bypassed, provided that there are no dependencies or that you do not have to force the object to the next state.

The life cycle of a feature begins when you open the feature (`open` state). As the originator, you can cancel a feature if it is in the `open` or `returned` state. You may decide to cancel a feature if it refers to an enhancement that has been implemented.

If you specified a release when you opened the feature, the feature moves from the `working` state to the `verify` state when the track for that release reaches the `complete` state. If you did not specify a release, the feature moves from the `working` state to the `verify` state when one of the tracks for that feature reaches the `complete` state.

The feature moves from the `verify` to the `closed` state when all verification records associated with the feature are marked and all tracks for the feature reach the `complete` state.

Displaying Feature Information in the Client Area

The CMVC - Features window allows you to open, modify and change the states of features.

When you query a feature, information is displayed in the client area of the window under the following headings:

Prefix Displays the prefix specified when the feature was opened. Your family administrator can configure a list of prefixes for your development environment.

Number Displays the alphanumeric identifier of the feature.

Component Displays the name of the component that manages the feature.

State Displays the current state of the feature.

Originator Displays the user ID of the feature originator.

Owner Displays the user ID of the feature owner.

Age Displays the time that has elapsed since the feature was opened.

Reference Displays an optionally assigned value that groups related features.

Abstract Displays a summary of the design that is proposed by the feature.

Priority Value that indicates the urgency for implementing the feature. Your family administrator can suppress this field or add new values.

Target A reference, such as a level or a date, for the work being done for the feature. You can use all characters except vertical bars (|), ASCII control characters, or shell metacharacters. You can type up to 15 alphanumeric characters.

Note: Other fields may also be displayed if your family administrator has configured additional fields. For more information, refer to “Working with Configurable Fields” on page 22 and the book *IBM CMVC Server Administration and Installation*, or see your family administrator.

Performing Queries with the Open Feature List

The Open Feature List dialog box allows you to perform queries and list information in the CMVC - Features window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to "Defining Default Queries and Tasks" on page 19.

To perform a query using the Open Feature List box:

1. Select `Open list...` from the File menu to display the Open Feature List dialog box.
2. Type the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

   Prefixes Prefixes specified when the feature was opened. Your family administrator can configure a list of prefixes for your development environment. Use the `Choices...` button to make your selection from a list of available alternatives.

   Features Identifier of the feature. You can type up to 15 alphanumeric characters.

   Components Component that manages the feature. For example, you want to list information about the features that are referenced by the `code` component.

   States Particular state or states of the feature. For example, you may want to list features that are in the `review` state.

   Originator IDs User ID of the feature originator. For example, you may want to list the features that `mikes` originated.

   Owner IDs User ID of the feature owners. For example, you may want to list the features that `patjones` owns.

   Ages Time that has elapsed since the feature was opened. Your family administrator can select the algorithm for computing the age.

   References Optionally assigned value that groups related defects or features.

   Abstracts Summary of the design change that the feature addresses. For example, you may want to list all features that have the keyword resource in them. You could do this by using the SQL wildcard character (`%`) and the `like` operator, and searching for the string `%resource%`.

   Duplicates Features that are specified by more than one feature or defect. For example, you may want to search for features that have the feature `1144` specified as a duplicate.

   Add dates Date (yy/mm/dd) on which the features were created. For example, you may want to list only the features created after `91/04/29`. Valid SQL operators for this field are `<`, `>`, `like`, `is null`, and `between`.

   Priority Value that indicates the urgency for implementing the feature. Your family administrator can suppress this field or add new values.

   Target A reference, such as a level or a date, for the work being done for the feature. You can use all characters except vertical bars (`|`), ASCII control characters, or shell metacharacters. You can type up to 15 alphanumeric characters.

3. Type the required information in any additional entry fields that your family administrator has configured.
4. Select `OK` to perform the query. The CMVC - Features window is refreshed, and the information that you requested is displayed in the client area.

You can use the `Refresh now` item on the File menu to reissue a query to the server. The most recent query is sent and the list is updated with current information from the server. If you issue a query to the server using the `Open list...` option, you can replace the query with the default query specified in the system file by selecting the `Default list` item from the File menu.

Note: For more information about how to use the Options menu items to specify operators, refer to "SQL Operator (in) and No Sort Combo Boxes" on page 11.

Performing Originator Actions with the CMVC - Features Window

This section describes the authority and prerequisites for performing common actions with the CMVC - Features window, and how to perform these actions.

Opening a Feature

You can open a feature to suggest a design change or an enhancement to a project.

Authority Required: None.

Prerequisites: When opening a feature, you must ensure that the following requirements are met:

- You must have a valid CMVC user ID for the CMVC family in which the enhancement needs to be addressed.
- You must know the name of the component that is affected by the suggested feature.

Procedure: To open a feature:

1. Select `Open...` from the Actions menu to display the Open Feature dialog box.
2. Type information in the following fields:

   Component Component that the feature addresses. By default, the owner of this component becomes the owner of the feature. You can import a selected component from the CMVC - Components window.

   Remarks A detailed description of the design you are proposing. An example is:

   ```
   Add a separate message catalog file for
   online help.
   ```

   You can type up to 15 999 alphanumeric characters.

   Abstract Summary of the proposed design. An example is:

   ```
   Add warning when message catalog not found.
   ```

Do not use a vertical bar (|), $, ", `, or \, because these have a special meaning to the shell. If you do not enter anything in this field, the default is to use the first 63 characters or the first line of information in the Remarks field. Typing in an abstract could be more meaningful than relying on the default.

Number Identifier of the feature. You can type up to 15 alphanumeric characters. If this field is left empty, CMVC assigns a number to the feature and informs you of the number.

Reference An optionally assigned value for a defect or feature that groups related defects and features. The reference name must not exceed 15 characters. You can use all characters except vertical bars (|), ASCII control characters or shell metacharacters.

Prefix Prefix that indicates the type of feature. Examples are s (suggestion made by customer) or f (feature requested by internal users). Your family administrator can configure a list of prefixes for your development environment. You can use the Choices... button to make your selection from a list of available alternatives.

3. Type the required information in any additional entry fields that have been configured by your family administrator.

   Note: For most of these fields, you can also use the Choices... button to make your selection from a list of alternatives.

4. Select OK to open the feature. An information box displays a message similar to the following:

   ```
   A new feature was opened successfully.
   The new feature number is 1444.
   ```

The value displayed is a unique identifier for the feature that you just opened. You can refresh the list in the CMVC - Features window to display the new feature and perform further actions on it.

The feature is in the open state and you become the originator of the feature. The owner of the component against which you opened the feature becomes the owner of the feature, and must perform one of the following actions:

- Move the feature to the design state if the feature design, size, and review (DSR) subprocess is specified for the associated component process.
- Accept the feature if the feature design, size, and review (DSR) subprocess is not specified for the associated component process.
- Return it to the originator.
- Assign it to a different component or user.

Canceling a Feature

You can cancel a feature if:

- The proposed design change cannot be implemented at this time.
- It is a duplicate of an existing feature or defect.

Authority Required: The feature originator has implicit authority. You have explicit authority if the FeatureCancel action is specified for your user ID in the component associated with the feature.

Prerequisites: When canceling a feature, you must ensure that the feature is in the `open` or `returned` state.

Procedure: To cancel a feature:

1. Select the feature you want to cancel from the list of features in the client area.
2. Select `Cancel...` from the Actions menu to display the Cancel Features dialog box. Accept the prefilled information in the `Features` field or type new information.
3. Type additional comments about the features you are canceling in the `Remarks` field. You can type up to 15 999 alphanumeric characters. An example is:

   ```
   I opened this feature prematurely.  I now think that
   it does not fit with the overall design of the product.
   ```

4. Select `OK` to cancel the feature. The feature moves to the `canceled` state. This is the final state of a invalid feature or a duplicate feature. No further action is required. You can reopen this feature, if needed.

Note: If you do not want to select the features from a list in the client area, you can begin with step 2 and type the information directly into the text entry fields.

Reopening a Feature

You can reopen a feature that was returned if you want the feature to be implemented. You can also reopen a feature that you canceled earlier if you want the changes to be implemented.

When you reopen a feature, you should:

- Provide details about the proposed design change
- Provide additional justification for implementing the feature.

Authority Required: The feature originator has implicit authority. You have explicit authority if the FeatureReopen action is specified for your user ID in the component associated with the feature.

Prerequisites: When reopening a feature, you must ensure that the feature is in the `canceled` or in the `returned` state.

Procedure: To reopen a feature:

1. Select the feature you want to reopen from the list of features in the client area.
2. Select `Reopen...` from the Actions menu to display the Reopen Features dialog box. Accept the prefilled information in the `Features` field or type new information.
3. Type information about the feature you are reopening in the `Remarks` field: You can type up to 15 999 alphanumeric characters. An example is:

   ```
   Timing is now right to implement these features,
   thanks to the new direction of the product.
   ```

4. Select `OK` to reopen the feature. The feature moves to the `open` state. The feature owner can design the feature or return it to the originator.

Note: If you do not want to select the features from a list in the client area, you can begin with step 2 and type the information directly into the text entry fields.

Verifying a Feature

You can verify that a feature has been implemented.

Authority Required: The feature owner has implicit authority. You have explicit authority if the FeatureVerify action is specified for your user ID in the component associated with the feature.

Prerequisites: The feature must be in the `working` state.

A feature with tracks moves from the `working` to the `verify` state automatically when the first track associated with the feature is complete. If no tracks exist, the feature owner must move the feature to the `verify` state when work is completed.

Procedure: To verify a feature:

1. Select the feature you want to force to the `verify` state from the list of features in the client area.
2. Select `Verify`... from the Actions menu to display the Verify Features dialog box. Accept the prefilled information in the `Features` field or type new information.
3. Type additional information, such as the reason you are moving the features from the `working` state to the `verify` state, in the `Remarks` field. You can type up to 15 999 alphanumeric characters. An example is:

   ```
   I forced this feature from working state to
   verify state because there are no tracks associated
   with this feature and I want the originator to verify
   the results using a verification record.
   ```

4. Select `OK` to move the feature to the `verify` state.

Existing verification records move from the `notReady` to the `ready` state. (There may be more than one verification record if duplicate defects or features exist. The verification records of those returned duplicates are activated by the movement of an active feature.)

If the verify subprocess is specified for the associated component, the owners of verification records must verify the outcome of the feature before the feature can move to the `closed` state. For more information on verification record owner tasks, see Chapter 6, "Working with Sizing Records and Verification Records" on page 83.

Note: If you do not want to select the features from a list in the client area, you can begin with step 2 and type the information directly into the text entry fields.

Adding Remarks to a Feature

You can add remarks or comments to a feature.

Authority Required: None.

Prerequisites: The feature exists in any state.

Procedure: To add remarks to a feature:

1. Select the feature to which you want to add remarks from the list of features in the client area of the CMVC - Features window.
2. Select `Add remarks...` from the Actions menu to display the Add Feature Remarks dialog box. Accept the prefilled information in the `Features` field or type new information.
3. Type your comments in the `Remarks` field. An example is:

   ```
   I heard that Joe Hamilton's development team
   implemented a similar feature.
   ```

   If you require additional space for your remarks, select `Edit...` to display the Remarks Editor dialog box. You can type up to 15 999 alphanumeric characters.
4. Select `OK` to save the remarks to the server. The remarks cannot be edited or removed.

Note: If you do not want to select the features from a list in the client area, you can begin with step 2 and type the information directly into the text entry fields.

Modifying the Originator of a Feature

You can assign the role of feature originator to another CMVC user ID.

Authority Required: The feature originator has implicit authority. You have explicit authority if the FeatureModify action is specified for your user ID in the component associated with the feature.

Prerequisites: When modifying the feature originator, you must ensure that the new originator has a valid CMVC user ID. The feature can exist in any state.

Procedure: To modify the feature originator:

1. Select the feature for which you want to specify a new originator.
2. Select `Originator...` from the Modify menu to display the Modify Feature Originator dialog box. Accept the prefilled information in the `Features` field or type new information.
3. Type information in the following fields:

   New originator User ID of the new originator of the specified features. You can also import a selected user ID from the CMVC - Users window.

   Remarks Additional comments about the modification. You can type up to 15 999 alphanumeric characters. An example is:

   ```
   I am going on vacation for 4 weeks and have
   given Paula Mathers originator responsibility
   over feature 1444.
   ```

4. Select `OK` to modify the originator. The new originator assumes responsibility for verifying the outcome of the feature. If you want to reassign the originator's verification record, you must do this manually.

Note: If you do not want to select the features from a list in the client area, you can begin with step 2 and type the information directly into the text entry fields.

Modifying the Properties of a Feature

As the feature originator, you can modify the following properties of a feature:

- Abstract
- Number
- Reference
- Prefix.

Authority Required: The feature originator and owner have implicit authority. You have explicit authority if the FeatureModify action is specified for your user ID in the component associated with the feature.

Prerequisites: The feature can exist in any state.

Procedure: To modify the properties:

1. Select the feature for which you want to modify properties from the list of features in the client area.
2. Select `Properties...` from the Modify menu to display the Modify Feature Properties dialog box. Accept the prefilled information in the `Features` field or type new information.
3. Type information in at least one of the following fields:

   Abstract Restatement of the summary of the proposed design. The abstract must not exceed 63 characters. An example is:

   ```
   Message catalog for online help.
   ```

   Do not use vertical bars (|), $, ", `, or \, because these have a special meaning to the shell.

   Number New identifier for the feature. You can type up to 15 alphanumeric characters.

   Reference Optionally assigned value for a feature that groups related features. You can type up to 15 alphanumeric characters. You can use all characters except vertical bars (|), ASCII control characters or shell metacharacters.

   Prefix Value that indicates the type of feature. These values may be configured by your family administrator. You can use the `Choices...` button to make your selection from a list of available alternatives.

   Priority Value that indicates the urgency for implementing the feature. Your family administrator can suppress this field or add new values.

   Target A reference, such as a level or a date, for the work being done for the feature. You can use all characters except vertical bars (|), ASCII control characters, or shell metacharacters. You can type up to 15 alphanumeric characters.

4. Type the required information in any additional entry fields that your family administrator has configured.
5. Type additional comments to indicate the reason for changing the properties in the `Remarks` field. You can type up to 15 999 alphanumeric characters. An example is:

```
The proposed design contained in feature 1444
needs to be expanded.  Work will cease until
more details are received.
```

6. Select `OK` to modify the properties. Any modifications that you make are reflected in the updated feature.

Note: If you do not want to select the features from a list in the client area, you can begin with step 2 and type the information directly into the text entry fields.

Showing the Details of a Feature

You can show the details of a feature, including the:

- Component to which the feature belongs
- Current state of the feature
- Add and change dates
- Names of the originator and the owner
- Tracks addressing the feature, if any
- Status of any existing verification records
- History of the actions made and remarks added to the feature.

Authority Required: The feature originator has implicit authority. You have explicit authority if the FeatureView action is specified for your user ID in the component associated with the feature.

Prerequisites: The feature can exist in any state.

Procedure: To show details for a single feature, double-click on a feature listed in the client area.

To show details for more than one feature&colon,

1. Select the features from the list of features in the client area.
2. Select `Details` from the Show menu. The CMVC - Information window is opened and the details are displayed.

Note: If you do not want to select features from a list in the client area, you can begin with step 2 and then type the information directly in the `Features` field.

Showing the Process Configurations of a Feature

You can show the configuration of the processes that are associated with the feature.

Authority Required: The feature owner has implicit authority. You have explicit authority if the FeatureView action is specified for your user ID in the component associated with the feature.

Prerequisites: The feature can be in any state.

Procedure: To show the process configurations for a feature:

1. Select the feature for which you want to show the process configurations from the list of features in the client area.
2. Select `Process configurations` from the Show menu. The CMVC - Information window is opened and the configured processes are displayed.

Note: If you do not want to select features from a list in the client area, you can begin with step 2 and then type the information directly in the `Features` field.

Showing the Change History of a Feature

You can show the history of changes for features. For more information about how to show the change history, refer to "Using the CMVC - Change History Window" on page 23.

Additional Show Options

Additional show options allow you to open certain GUI windows, including the CMVC - Sizing Records, CMVC - Verification Records, CMVC - Tracks, CMVC - Approval Records, CMVC - Fix Records, CMVC - Test Records, and CMVC - Level Members windows. To open these windows, select the corresponding option from the Show menu.

Feature Owner's Tasks

A feature owner is responsible for entering design details, sizing the resources needed to implement the feature, reviewing the feasibility of the proposed feature, and deciding whether to accept a feature for implementation or to return it to the originator.

You can assign some of these tasks to other users in your work group. You can also assign the feature to another owner if you believe that the feature does not apply to you.

Some of the actions that you can perform as a feature owner include:

- Design, size, and review a feature
- Accept or return a feature
- Reassign a feature to another component
- Create a track
- Modify the properties or the owner of a feature
- Show the process configurations for a feature.

For a description of how to modify the properties of a feature and how to show the process configurations, refer to "Feature Originator's Tasks" on page 65.

Performing Owner Actions with the CMVC - Features Window

This section describes the authority and prerequisites for performing common actions with the CMVC - Features window, and how to perform these actions.

Designing a Feature

You can move a feature to the `design` state to add specific design details.

Authority Required: The feature owner has implicit authority. You have explicit authority if the FeatureDesign action is specified for your user ID in the component associated with the feature.

Prerequisites: The feature can be in either the `open`, `design`, `size`, `review`, or `returned` state.

Procedure: To design a feature:

1. Select the feature that you want to move to the `design` state from the list of features in the client area.
2. Select `Design...` from the Actions menu to display the Design Features dialog box. Accept the prefilled information in the `Features` field or type new information.
3. Type additional comments, such as the design text, in the `Remarks` field. You can type up to 15 999 alphanumeric characters. An example is:

   ```
   Reorder the fields in the Compose Music
   dialog box.
   ```

4. Select `OK` to move the features to the `design` state . This occurs if the feature was in any of the `open`, `returned`, `size`, or `review` states.
5. If you moved a feature from the `review` state to the `design` state, you will have to remark the sizing records.

You can move a feature to:

- The `design` state so that you can enter additional design information as often as required
- The `size` state to indicate the resources needed to implement the design
- The `returned` state if you feel that, after all, the feature should not (or cannot) be implemented.

Note: If you do not want to select the features from a list in the client area, you can begin with step 2 and type the information directly into the text entry fields.

Sizing a Feature

You can move the feature to the `size` state so that the effort to implement features can be sized.

Authority Required: The feature owner has implicit authority. You have explicit authority if the FeatureSize action is specified for your user ID in the component associated with the feature.

Prerequisites: The feature must be in the `design` state with design details entered as text.

Procedure: To size a feature:

1. Select the feature you want to size from the list of features in the client area.
2. Select `Size`... from the Actions menu to display the Size Features dialog box. Accept the prefilled information in the `Features` field or type new information.
3. Type additional information, such as sizing information, in the `Remarks` field. You can type up to 15 999 alphanumeric characters. An example is:

   ```
   Shelley, ask Peter to help you with the estimates.
   ```

4. Select `OK` to move the feature to the `size` state. Any sizing records that exist for that feature will move to the `ready` state. You can create additional sizing records.

Note: If you do not want to select the features from a list in the client area, you can begin with step 2 and type the information directly into the text entry fields.

Reviewing a Feature

You can move a feature to the `review` state from the `size` state so that a formal review of the proposed design implementation and its sizing information can take place.

Authority Required: The feature owner has implicit authority. You have explicit authority if the FeatureReview action is specified for your user ID in the component associated with the feature.

Prerequisites: All sizing records must be marked.

Procedure: To review a feature:

1. Select the feature for which you want the design and sizing information formally reviewed.
2. Select `Review...` from the Actions menu to display the Review Features dialog box. Accept the prefilled information in the `Features` field or type new information.
3. Type additional information about the features that you want reviewed in the `Remarks` field. You can type up to 15 999 alphanumeric characters. An example is:

   ```
   Keep in mind that two programmers are on holidays
   next month.
   ```
4. Select `OK` to review the feature.

If you have marked all sizing records, then you can move a feature to the `review` state. You can return the feature from the `review` state to the originator if you do not want to implement it. You can accept the feature for implementation and thereby change the state of the feature to `working`. You can move a feature from the `review` state to the `design` state if additional design information is required. If you do so, you will have to remark the sizing records.

Note: If you do not want to select the features from a list in the client area, you can begin with step 2 and type the information directly into the text entry fields.

Accepting a Feature for Implementation

You can accept a feature so that it moves to the `working` state.

Authority Required: The feature owner has implicit authority. You have explicit authority if the FeatureAccept action is specified for your user ID in the component associated with the feature.

Prerequisites: When accepting a feature for implementation, you must ensure that:

- If the feature design, size, and review (DSR) subprocess is included in the process for the component, the feature is in the `open` or `returned` state.
- If the feature design, size, and review (DSR) subprocess is not included in the process for the component, the feature is in the `open` state.

Procedure: To accept a feature:

1. Select the feature you want to implement from the list of features in the client area.
2. Select `Accept...` from the Actions menu to display the Accept Features dialog box. Accept the prefilled information in the `Features` field or type new information.
3. Type additional information about the features you want implemented in the `Remarks` field. You can type up to 15 999 alphanumeric characters. An example is:

   ```
   Send the feature to Mary Ferguson for
   implementation.
   ```
4. Select `OK` to implement the feature. The feature moves to the `working` state.

At this time, one track is created for every release that has a sizing record in the `accept` state. Additional tracks can be created for releases for which the tracking subprocess is configured. (See "Creating a Track" on page 78.) From the `working` state, feature changes follow the same process as defects. A verification record is created in the `notReady` state for the feature originator.

Note: If you do not want to select the features from a list in the client area, you can begin with step 2 and type the information directly into the text entry fields.

Creating a Track

Tracks are required for features for which the tracking subprocess is included in the process for the release. A track is created automatically in features for each sizing record that is in the `accept` state. Otherwise, a track has to be created manually. The track life cycle begins when the feature is in the `working` state.

A track does not have a single name. Rather, a track is referred to by specifying a release name and a feature name.

Authority Required: The feature owner has implicit authority. You have explicit authority if the TrackCreate action is specified for your user ID in the component associated with the feature.

Prerequisites: When creating tracks, you must ensure that the following requirements are met:

- The process of the release for which you want to create a track must have the tracking subprocess specified.
- The feature must be in either the `working` or the `verify` state.

Procedure: To create a track:

1. Select the feature for which you want to create a track from the list of features in the client area. Accept the prefilled information in the `Features` field or type new information.
2. Select `Create tracks...` from the Actions menu to display the Create Tracks dialog box.
3. Type information in the following fields:

Features Names of existing features to be tracked.

Releases Names of existing releases in which changes will be tracked to implement the features. You can also import selected releases from the CMVC - Releases window.

Target A reference, such as a level or a date, for the work being done for the tracks. You can use all characters except vertical bars (|), ASCII control characters or shell metacharacters. You can type up to 15 alphanumeric characters.

Owner User ID of the track owner. If you do not specify an owner, then the default is the user creating the track. You can import a selected user ID from the CMVC - Users window.

4. Select `OK` to create the track.

Note: If you do not want to select features from a list in the client area, you can begin with step 2 and then type the information directly in the `Features` field.

When the feature is accepted, tracks and fix records are created for all sizing records that are marked as `accept`.

You can create additional tracks if you find that the same feature applies to other releases. For example, feature `333` was opened to add a function to `release4`. After further investigation you may discover that other releases should also have the feature implemented.

Returning a Feature

You can return a feature to the originator if a suggested enhancement is not necessary, is impossible to implement at this time, or is a duplicate of an existing feature or defect.

Authority Required: The feature owner has implicit authority. You have explicit authority if the FeatureReturn action is specified for your user ID in the component associated with the feature.

Prerequisites: When returning a feature, you must ensure that the following requirements are met:

- Feature is in one of the `open`, `design`, `size`, or `review` states.
- If the feature is in the `working` state, there must be no tracks associated with that feature.

You can also return a feature to the originator if the feature is a duplicate of an existing feature or defect.

Procedure: To return a feature:

1. Select the feature you want to return from the list of features in the client area.
2. Select `Return...` from the Actions menu to display the Return Features dialog box. Accept the prefilled information in the `Features` field or type new information.
3. If you are returning the feature because it is a duplicate of an existing feature or defect, identify the duplicate by typing the alphanumeric identifier in the Duplicate field.

4. Type additional information about the returned features in the `Remarks` field. You can type up to 15 999 alphanumeric characters. An example is:

   ```
   The work needed to implement this feature is too
   great in comparison to the expected improvement
   in function.
   ```

5. Select `OK` to return the feature.

The feature moves to the `returned` state and the originator must address it. The originator can:

- Reopen it to the same component, with additional design information or justification
- Reopen it to another component that is more appropriate for implementing the feature
- Cancel it if it no longer has to be implemented.

If you change your mind, you can accept a feature while it is in the `returned` state provided that the feature design, size, and review (DSR) subprocess is not included in the process for the associated component.

Note: If you do not want to select the features from a list in the client area, you can begin with step 2 and type the information directly into the text entry fields.

Modifying the Owner of a Feature

You can assign a feature to another CMVC user ID, thus transferring the responsibility of the feature ownership.

Authority Required: The feature owner has implicit authority. You have explicit authority if the FeatureAssign action is specified for your user ID in the component associated with the feature.

Prerequisites: The feature exists in any state.

Procedure: To modify ownership of a feature:

1. Select the feature for which you want to modify the owner from the list of features in the client area.
2. Select `Owner...` from the Modify menu to display the Modify Feature Owner dialog box. Accept the prefilled information in the `Features` field or type new information.
3. Type information in the following fields:

   New owner User ID of the person who will be the new owner of the specified features. You can also import a selected user ID from the CMVC - Users window.

   Remarks Additional information relating to the features. You can type up to 15 999 alphanumeric characters. An example is:

   ```
   Chris: I have assigned all of the features that I
   own to you for the duration of my educational leave.
   ```

4. Select `OK` to modify the feature owner. The new owner assumes the responsibilities of the feature owner.

Note: If you do not want to select features from a list in the client area, you can begin with step 2 and then type the information directly in the `Features` field.

Reassigning a Feature to Another Component

If a feature is opened against a component, but the feature actually affects another component, you can reassign the feature to the other component.

Authority Required: The feature owner and originator have implicit authority. You have explicit authority if the FeatureAssign action is specified for your user ID in the component associated with the feature.

Prerequisites: When reassigning a feature to another component, you must ensure that the following requirements are met:

- A feature originator can specify a new component for the feature when the feature is in the `open` state only.
- The feature owner can specify a new component regardless of the state of the feature.

Procedure: To reassign a feature to another component:

1. Select the feature for which you want to specify a new component from the list of features in the client area.
2. Select `Component...` from the Modify menu to display the Modify Feature Component dialog box. Accept the prefilled information in the `Features` field or type new information.
3. Type information in the following fields:

   New component Name of the component to which you want to reassign the features. You also can import a selected component from the CMVC - Components window.

   New owner User ID of the person who will be the new owner of the selected features. You also can import a selected User ID from the CMVC - Users window. Information for this field is optional.

   Remarks Additional information relating to the modification. You can type up to 15 999 characters. An example is:

   ```
   I am reassigning feature 1444 to the
   component that manages message catalogs.
   ```
4. Select `OK` to modify the component. The feature is associated with the new component and the new component owner or other person you specify assumes the responsibilities of the feature owner.

Note: If you do not want to select features from a list in the client area, you can begin with step 2 and then type the information directly in the `Features` field.

Chapter 6. Working with Sizing Records and Verification Records

You can use the CMVC - Sizing Records window and the CMVC - Verification Record window to perform actions on sizing and verification records. This chapter discusses the actions that you can perform as a sizing record owner and a verification record owner. A superuser can perform any of the actions described in this chapter.

Sizing Record Owner's Tasks

Sizing records can be created when the defect or feature is in the `size` state. Each sizing record represents the resources needed to address the defect or to implement the feature for one release in one component. Therefore, if defects or features are required in multiple releases and components, the number of sizing records increases.

By default, the owner of the component is also the owner of the sizing record created for that component. The sizing record owner is responsible for entering information about the approximate amount of work required to implement the proposed feature.

Some of the actions that you can perform as a sizing record owner include:

- Creating, accepting, rejecting, or deleting a sizing record
- Modifying a sizing record
- Showing the details of a sizing record
- Showing the note history for a sizing record.

Displaying Sizing Information in the Client Area

The CMVC - Sizing Records window allows you to create, delete, accept, reject, and modify sizing records.

When you query a sizing record, specific information is displayed in the client area of the window under the following headings:

Prefix Displays the value specified when the defect or feature was opened.

Number Displays the identifier of the defect or feature to which the sizing records refers.

Component Displays the name of the component for which the sizing is to be done.

Release Displays the release in which the work that addresses the particular defect or feature is being performed.

State Displays the current state of the sizing record. Valid states are ready, accept, or reject.

Owner Displays the user ID of the owner of the sizing record.

Reference Displays an optionally assigned value that groups related defects or features.

Created Displays the date and time (yy/mm/dd hh:mm:ss) that the sizing record was created.

Updated Displays the date and time (yy/mm/dd hh:mm:ss) that the sizing record was last updated.

Abstract Displays a summary of the defect or feature to which the sizing record refers.

Performing Queries with the Open Sizing Record List

The Open Sizing Record List dialog box allows you to perform queries and list information in the CMVC - Sizing Records window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to "Defining Default Queries and Tasks" on page 19.

To perform a query using the Open Sizing Record List dialog box:

1. Select `Open list...` from the File menu to display the Open Sizing Record List dialog box.
2. Type the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

   Prefixes Prefixes specified when the defects or features were opened that describe the type of defect or feature. Your family administrator can configure these values. Use the `Choices...` button to select from a list of these values.

   Defects/Features Identifiers of the defects or features. For example, you may want to list information about specific defects or features.

   Components Components to which the sizing records refer. For example, you may want to list sizing records that are referenced by the `code` component.

   Releases Releases for which the sizing records exist. For example, you may want to list information about sizing records that are associated with the `debugger2` release.

   States Particular state or states of the sizing records. Valid options are `ready`, `accept`, or `reject`. For example, you may want to list sizing records that are in the `accept` state.

   Owner IDs User IDs of the owners of the sizing records. For example you may want to list the sizing records that `patjones` owns.

   References Optionally assigned value that is used to group related defects or features.

   Add dates Dates (yy/mm/dd) on which the sizing records were created. For example, you may want to list only those sizing records created after `93/12/29`. Valid SQL operators for this field are <, >, like, is null, and between.

Change dates Dates (yy/mm/dd) on which the sizing records were last updated. For example, you may want to list only those sizing records updated after 93/12/01. Valid SQL operators for this field are <, >, like, is null, and between.

Abstracts Summary of the change that the defect or feature was opened to address. For example, you may want to list all the sizing records for features that have the keyword resource in them. You can do this by using the SQL wildcard character (%) along with the like operator, and search for the string %resource%.

Sizing Information Summary of the resources that are required to resolve a defect or implement a feature. For example, you may want to list all the sizing records that indicate that the implementation of a feature will require 10 hours of work. You can do this by using the wildcard character (%) along with the SQL like operator and searching for the string %10 hours%.

3. Select OK to perform the query. The CMVC - Sizing Records window is refreshed, and the information that you requested is displayed in the client area.

You can use the Refresh now item on the File menu to reissue a query to the server. The most recent query is sent and the list is updated with current information from the server. If you issue a query to the server using the Open list... option, you can replace the query with the default query specified in the system file by selecting the Default list item from the File menu.

Note: For more information about how to use the Options menu items to specify operators, refer to “SQL Operator (in) and No Sort Combo Boxes” on page 11.

Performing Actions with the CMVC - Sizing Records Window

This section describes the authority and prerequisites for performing common actions with the CMVC - Sizing Records window and how to perform these actions.

Creating a Sizing Record

You can create sizing records so that specific users can estimate the effort required to resolve the defect or implement the feature.

Authority Required: The defect or feature owner has implicit authority. You have explicit authority if the SizeCreate action is specified for your user ID in the component associated with the defect or feature.

Prerequisites: The defect or feature you want to size is in the size state.

Procedure: To create sizing records:

1. Select Create... from the Actions menu to display the Create Sizing Records dialog box.

2. Type information in the following text entry fields:

   Defects/Features Identifiers of the defects or features for which you want to create sizing records. You can type up to 15 alphanumeric characters. You can also import selected defects or features from the CMVC - Defects or the CMVC - Features window.

   Components Name of the components for which you want to create sizing records. You can also import selected components from the CMVC - Components window.

   Release Name of the release in which the defects or features will be implemented. You can also import a selected release from the CMVC - Releases window.

3. Select `OK` to create the sizing records.

Sizing records are created and each sizing record is owned by the owner of the component for which it was created. The sizing records are created in the `ready` state and the sizing record owners must address them.

Once all sizing records are marked by sizing record owners, the defect or feature owner or someone with appropriate authority can move the defect or feature to the `review` state. If sizing records have to be remarked because the defect or feature went from the `review` state to the `design` state, then the sizing records are reset to the `ready` state.

Accepting a Sizing Record

You accept sizing records to indicate the resources that are required to resolve a defect or implement a feature in the specified components and releases.

Note: Each sizing record refers to one component - release pair.

Authority Required: The owner of the sizing record has implicit authority. You have explicit authority if the SizeAccept action is specified for your user ID in the component associated with the sizing record.

Prerequisites: When accepting a sizing record, you should ensure that the following requirements are met:

- Defect or feature is in the `size` state.
- Sizing records can exist in any state.

Procedure: To accept a sizing record:

1. Select the sizing record you want to accept from the list of records in the client area.
2. Select `Accept...` from the Actions menu. An Accept Sizing Records dialog box is displayed.
3. In the `Sizing` field, type the amount of time or resources required to implement the proposed feature or correct the defect in the specified release and components. Do not use `$`, `"`, `'`, or `\` because these have a special meaning to the shell. The maximum length is 63 characters.
4. Select `OK` to accept the sizing record.

A track is created for releases for which the tracking subprocess is specified, when the defect or feature moves to the `working` state. One track is created for each sizing record that is in the `accept` state.

When all sizing records have been accepted or rejected, the defect or feature is ready to be reviewed by the defect or feature owner or by someone with proper authority.

Note: If you do not want to select sizing records from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Rejecting a Sizing Record

You can reject sizing records to indicate that additional resources are not required to resolve a defect or implement the feature in the specified components and releases.

Note: Each sizing record refers to one component - release pair.

Authority Required: The owner of the sizing record has implicit authority. You have explicit authority if the SizeReject action is specified for your user ID in the component associated with the sizing record.

Prerequisites: When rejecting a sizing record, you must ensure that the following requirements are met:

- Defect or feature is in the `size` state
- Sizing records exist in CMVC.

Procedure: To reject a sizing record:

1. Select the sizing record you want to reject from the list of sizing records in the client area.
2. Select `Reject...` from the Actions menu. A Reject Sizing Records dialog appears. Accept the prefilled information in the `Defects/Features`, `Components`, and `Release` fields, or type in new information.
3. Select `OK` to reject the sizing record.

When all sizing records have been accepted or rejected, the defect or feature is ready to be reviewed by the defect or feature owner or by someone with proper authority.

Note: If you do not want to select sizing records from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Deleting a Sizing Record

You can delete sizing records if you no longer need an estimate of the effort required to resolve a defect or to implement a feature.

Authority Required: The defect or feature owner has implicit authority. You have explicit authority if the SizeDelete action is specified for your user ID in the component associated with the defect or feature.

Prerequisites: When deleting sizing records, you must ensure that the following requirements are met:

- Defect or feature is in the `size` state
- Sizing record can exist in any state.

Procedure: To delete sizing records:

1. Select the sizing record that you want to delete.
2. Select `Delete...` from the Actions menu. A Delete Sizing Records dialog appears. Accept the prefilled information in the `Defects/Features`, `Components`, and `Release` fields, or type in new information.
3. Select `OK` to delete the sizing record.

If this was the last sizing record to be marked for the defect or feature, then the defect or feature owner or someone with appropriate authority can now move the defect or feature to the `review` state.

Note: If you do not want to select sizing records from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Reassigning a Sizing Record

You can assign the role of sizing record owner to another user, thus making them responsible for sizing the resources required to implement a given defect or feature in a given component - release pair.

Authority Required: The owner of the sizing record has implicit authority. You have explicit authority if the SizeAssign action is specified for your user ID in the component associated with the sizing record.

Prerequisites: When reassigning a sizing record, you must ensure that the following requirements are met:

- Defect or feature is in the `size` state
- Sizing record exists in CMVC
- New owner has a valid CMVC user ID.

Procedure: To reassign a sizing record:

1. Select the sizing record you want to assign to a new owner from the list of records in the client area.
2. Select `Owner...` from the Modify menu to display the Modify Sizing Record Owner dialog box. Accept the prefilled information in the `Defects/Features`, `Components`, and `Release` fields, or type in new information.
3. Type the user ID of the new owner of the sizing records in the `New owner` field. You can also import a selected user ID from the CMVC - Users window.
4. Select `OK` to modify the owner. The new sizing record owner must size the defect or feature and mark the sizing record when it moves to the `ready` state. To do this, the owner must either accept or reject the sizing record.

Note: If you do not want to select sizing records from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Showing the Details of Sizing Records

You can show the following details of sizing records:

- State of the sizing record
- Sizing information, if the sizing record is in the `accept` state
- Owner of the sizing record
- Summary of the defect or feature that the sizing record addresses.

Authority Required: None.

Prerequisites The sizing record exists.

Procedure: To show details for a sizing record, double-click on a sizing record listed in the client area of the CMVC - Sizing Records window.

Another way to show details of a sizing record is:

1. Select the sizing record for which you want to show details from the list of sizing records in the client area of the CMVC - Sizing Records window.
2. Select `Details` from the Show menu. The CMVC - Information window is opened and the details of the sizing records are displayed.

Note: If you do not want to select sizing records from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Showing the Note History of Sizing Records

You can show the note history of the sizing records that are associated with a defect or feature.

Authority Required: None.

Prerequisites: The sizing record exists.

Procedure: To show the note history of a sizing record:

1. Select the sizing record for which you want to see the note history of the associated defects or features from the list of records in the client area of the CMVC - Sizing Records window.
2. Select `Note history` from the Show menu. The CMVC - Information window is opened and the notes that refer to the defect or feature are displayed.

Note: If you do not want to select sizing records from a list in the client area, you can begin with step 2 and type the information directly into the text entry field.

Verification Record Owner's Tasks

When a defect or feature is accepted, that is, moved to the `working` state, a verification record is created for the originator to ensure that the feature was implemented, or that the defect was resolved to the originator's satisfaction.

The originator can reassign the ownership of the verification record to someone else, thus transferring the responsibility for verifying the outcome of the defect or feature.

The verification record is created in the `notReady` state but moves to the `ready` state when the defect or feature moves to the `verify` state.

Once the owner of the verification record accepts, abstains, or rejects the outcome of the defect or feature, the defect or feature can be closed.

The verification record owner can perform the following tasks:

- Accept a verification record
- Reject a verification record
- Abstain on a verification record
- Modify the ownership of a verification record
- Show the note history of a verification record.

Note: Verification records may not be required in your development environment. For more information about whether verification records are used, see your family administrator.

Displaying Verification Record Information in the Client Area

The CMVC - Verification Records window allows you to record the results of your defect or feature verification. When you query a verification record, information is displayed in the client area of the window under the following headings:

Prefix Displays the prefix specified when the defect or feature was opened. Your family administrator can configure a list of prefixes for your development environment.

Number Displays the identifier of the defect or feature that the verification record addresses.

Owner Displays the user ID of the owner of the verification record.

Component Displays the name of the component associated with the verification record.

State Displays the current state of the verification record. Valid states are `notReady`, `ready`, `accept`, `abstain`, or `reject`.

Type Indicates whether the verification record belongs to the original defect or feature, or to a duplicate defect or feature.

Reference Displays an optionally assigned value that groups related defects and features.

Created Displays the date and time (yy/mm/dd hh:mm:ss) on which the verification record was created.

Updated Displays the date and time (yy/mm/dd hh:mm:ss) on which the verification record was updated.

Abstract Displays a summary of the defect or feature that the verification record addresses.

Performing Queries with the Open Verification Record List

The Open Verification Record List dialog box allows you to perform queries and list information in the CMVC - Verification Records window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to "Defining Default Queries and Tasks" on page 19.

To perform a query using the Open Verification Record List dialog box:

1. Select `Open list...` from the File menu to display the Open Verification Record List dialog box.
2. Enter the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

   Prefixes Prefixes specified when the defect or feature was opened. Your family administrator can configure a list of choices for your development environment. You can use the `Choices...` button to select from a list of available alternatives.

   Defects/Features Identifier of the defects or features that the record addresses. You can type up to 15 alphanumeric characters in this field.

   Owner IDs User ID of the verification record owner. For example, you may want to see the verification records that `patjones` owns.

   Components Components to which the verification records refer. For example, you may want to list the verification records that are referenced by the `code` component.

   States Specific state of the verification records. Valid options are `notReady`, `ready`, `accept`, `abstain`, and `reject`. For example, you may want to list verification records that are in the `ready` state.

   Types Indicates whether the verification record belongs to an original defect or feature, or to a duplicate defect or feature. For example, you may want to see verification records for defects that are duplicates of other defects.

   References Optionally assigned values for defects or features that are used to group related defects and features.

   Add dates Dates (yy/mm/dd) on which the verification records were created. For example, you may want to list only those verification records created after `93/12/29`. Valid SQL operators for the field are `<`, `>`, `like`, `is null`, and `between`.

Change dates Dates (yy/mm/dd) on which the verification records were last changed. For example, you may want to list only those verification records changed after 93/12/29. Valid SQL operators for the field are <, >, like, is null, and between.

Abstracts Summaries of the defects or features that the verification records address. For example, you may want to list all verification records for defects or features that have the product name alpha in the abstract. You could do this by searching for the string %alpha% using the like SQL operator.

3. Select OK to perform the query. The CMVC - Verification Records window is refreshed, and the information that you requested is displayed in the client area.

You can use the Refresh now item on the File menu to reissue a query to the server. The most recent query is sent and the list is updated with current information from the server. If you issue a query to the server using the Open list... option, you can replace the query with the default query specified in the system file by selecting the Default list item from the File menu.

Note: For more information about how to use the Options menu items to specify operators, refer to "SQL Operator (in) and No Sort Combo Boxes" on page 11.

Performing Actions with the CMVC - Verification Records Window

This section describes the authority and prerequisites for performing common actions with the CMVC - Verification Records window and how to perform these actions.

Accepting a Verification Record

You can accept a verification record to indicate that you are satisfied with the way in which a defect was resolved or a feature was implemented.

Authority Required: The owner of the verification record has implicit authority. You have explicit authority if the VerifyAccept action is specified for your user ID in the component associated with the defect or feature.

Prerequisites: When accepting a verification record, you should ensure that the following requirements are met:

- Defect or feature is in the verify state
- Verification record is in the ready state.

Procedure: To accept a verification record:

1. Select the verification record you want to accept from the list of records in the client area.
2. Select Accept... from the Actions menu. An Accept Verification Records dialog box is displayed. Accept the prefilled information in the Defects/Features and Owner fields or type in new information.
3. Select OK to accept the verification record. The verification record moves to the accept state.

When all verification records are marked (even if one or more of them is marked reject or abstain) and all tracks are complete, the associated defect or feature moves to the `closed` state.

If a defect or feature is in the `closed` state, it cannot be reopened.

Note: If you do not want to select verification records from a list in the client area, you can begin with step 2 and then type the information directly in the text entry fields.

Rejecting a Verification Record

You can reject a verification record to indicate that you are not satisfied with the way in which a defect was resolved or a feature was implemented.

Authority Required: The owner of the verification record has implicit authority. You have explicit authority if the VerifyReject action is specified for your user ID in the component associated with the defect or feature.

Prerequisites: When rejecting a verification record, you should ensure that the following requirements are met:

- Defect or feature is in the `verify` state
- Verification record must be in the `ready` state.

Procedure: To reject a verification record:

1. Select the verification record you want to reject from the list of records in the client area.
2. Select `Reject...` from the Actions menu. A Reject Verification Records dialog box is displayed. Accept the prefilled information in the `Defects/Features` and `Owner` fields or type in new information.
3. Select `OK` to reject the verification record. The verification record moves to the `reject` state.

When all verification records are marked (even if one or more of them is marked `reject` or `abstain`) and all tracks are complete, the associated defect or feature moves to the `closed` state.

The defect or feature moves to the `closed` state and cannot be reopened. You can, however, open a new defect or feature, referencing the original defect or feature, if required.

Note: If you do not want to select verification records from a list in the client area, you can begin with step 2 and then type the information directly in the text entry fields.

Abstaining from Verification

You can abstain from verifying the outcome of the defect or feature associated with the verification record if:

- You want the defect or feature to be addressed differently.
- You are unable to assess the outcome of the defect or feature.

Authority Required: The owner of the verification record has implicit authority. You have explicit authority if the VerifyAbstain action is specified for your user ID in the component associated with the defect or feature.

Prerequisites: When abstaining from verification, you should ensure that the following requirements are met:

- Defect or feature is in the verify state
- Verification record is in the `ready` state

Procedure: To abstain from verification:

1. Select the verification records you want to mark with abstain from the list of records in the client area.
2. Select `Abstain...` from the Actions menu. An Abstain From Verification dialog box is displayed. Accept the prefilled information in the `Defects/Features` and `Owner` fields or type in new information.
3. Select `OK` to abstain from verifying the verification record. The verification record moves to the `abstain` state.

When all verification records are marked (even if one or more of them is marked `reject` or `abstain`) and all tracks are completed, the associated defect or feature moves to the `closed` state. The defect or feature moves to the `closed` state and cannot be reopened.

Note: If you do not want to select verification records from a list in the client area, you can begin with step 2 and then type the information directly in the text entry fields.

Modifying the Verification Record Owner

You can assign the role of verification record owner to another CMVC user ID.

Authority Required: The owner of the verification record has implicit authority. You have explicit authority if the VerifyAssign action is specified for your user ID in the component associated with the defect or feature.

Prerequisites: When modifying the verification record owner, you should ensure that the following requirements are met:

- Defect or feature is in the `working` or `verify` state
- Verification record is in the `ready` or `notReady` state
- New owner has a valid CMVC user ID.

Procedure: To modify the owner of the verification record:

1. Select the verification record for which you want to modify the owner from the list of records in the client area.
2. Select `Owner...` from the Modify menu to display the Modify Verification Record Owner dialog box. A Modify Verification Record Owner dialog box is displayed. Accept the prefilled information in the `Defects/Features` and `Owner` fields, or type in new information.
3. Type the user ID of the person you want to be the new owner of the verification record in the `New owner` field. You can also import a selected user ID from the CMVC - Users window.
4. Select `OK` to modify the owner. The new verification record owner assumes responsibility for verifying the outcome of the defect or feature associated with the verification record. The owner can accept, abstain, or reject the verification record.

Note: If you do not want to select verification records from a list in the client area, you can begin with step 2 and then type the information directly in the text entry fields.

Showing the Note History of Verification Records

You can show the note history of the verification records that are associated with a defect or feature.

Authority Required: None.

Prerequisites: The verification records exist in any state.

Procedure: To show details for a single verification record, double-click on a verification record listed in the client area.

To show details for more than one verification record:

1. Select the verification records for which you want to show the note history from the list of records in the client area of the CMVC - Verification Records owner window.
2. Select `Note history` from the Show menu. The CMVC - Information window is opened and all notes that refer to the defects or features are displayed.

Note: If you do not want to select verification records from a list in the client area, you can begin with step 2 and then type the information directly into the Defects/Features field in the Note History dialog box.

Chapter 7. Working with Tracks

The CMVC - Tracks window allows you to perform a variety of actions on tracks. This chapter describes the actions that you can perform as a track owner. If you are a superuser, you can perform any tasks that are described in this chapter.

Track Owner's Tasks

Tracks may be created by a defect or feature owner to monitor the work in a specific release. Multiple tracks indicate that the defect or feature needs to be addressed in more than one release. For example, a feature that is proposed for a software application (Release `prog1rel2`) will have to be documented (Release `doc1rel2`). Only one feature needs to be opened, and two tracks can be created to follow the work toward that feature in each release.

The user who creates the track is the track owner by default, unless track ownership is reassigned.

Some of the actions that you can perform as a track owner include:

- Cancel a track
- Move a track from the `fix` state to the `integrate` state
- Move a track from the `integrate` state to the `fix` state
- Move a track from the `integrate` state to the `commit` state
- Move a track from the `commit` state to the `test` state
- Move a track from the `test` state to the `complete` state
- Identify and remove corequisite tracks
- Check a track for dependencies
- Modify ownership or the target of a track
- Show the track details, note history, level members, and change history.

Note: The role of track owner exists only if the tracking subprocess is included in the associated release process.

Releases following a process that includes the tracking subprocess can also include the level, fix, approval, and test subprocesses. If one of these subprocesses is not included in the associated release process, the state associated with the subprocess is bypassed if there are no dependencies. For further information about configurable processes, refer to "Working with Configurable Processes" on page 23 or the book *IBM CMVC Server Administration and Installation*.

States of a Track

You can create tracks to monitor the progress of the changes to either resolve a defect or implement a feature within a release for which the tracking subprocess is included. A track is referred to by specifying either an association between a release and a defect or an association between a release and a feature (a track does not have its own identifier).

The control that tracks have over changes is reflected in the various states a track goes through and the way in which a track moves from state to state. Normally, all changes in states are automatic. Changes occur when certain actions are performed on CMVC objects, such as, a level, or approval, fix, and test records.

However, you can force a track to change states if you have been granted sufficient authority.

The following list summarizes how tracks change states:

- A track moves from the `fix` state to the `integrate` state when all fix records are completed. Configurable processes may allow you to move the track from the `fix` state to the `commit` state, or from the `fix` state to the `complete` state.
- A track can be included as a level member in a level when the track is in the `integrate` state, and when the level subprocess has been included for the release process associated with the track. The track moves to the `commit` state when the level is committed. You can force the track to the `commit` state without committing the level. For example, you may do this if you decide not to make changes and you want to push the track through the states to close the defect or feature that the track addresses. You can force the track from the `integrate` state to the `commit` state only if there are no file changes that are associated with the track, or if the level subprocess is not included in the associated process for the release.
- A track moves from the `commit` to the `test` state when the level in which it is included moves to the `complete` state. You can force the track from the `commit` state to the `test` state, if the track is ready for testing. You can also force the track from the `commit` state to the `test` state, for example, if you do not want to make changes and want to push the track through the states so that you can close the defect or feature that the track addresses.
- A track moves from the `commit` state to the `complete` state, if the test subprocess is not included in the associated process for the release. You may want to force the track from the `test` state to the `complete` state, provided that there are no file changes associated with the track. If there are file changes associated with the track, test records must be marked. Once they are completed, the track moves from the `test` state to the `complete` state.

For more information about the states of tracks, refer to the book *IBM CMVC Concepts*.

Displaying Track Information in the Client Area

The CMVC - Tracks window allows you to work with tracks in CMVC. When you query a track, specific information is displayed in the client area of the window under the following headings:

Prefix Displays the prefix specified when the defect or feature was opened. Your family administrator can configure a list of choices for your development environment.

Number Displays the alphanumeric identifier of the feature or defect that the track addresses.

Release Displays the release that the track for a particular defect or feature addresses.

State Displays the current state of the track. Valid states are `approve`, `fix`, `integrate`, `commit`, `test`, or `complete`.

Target Displays a target, such as a level or date, for performing the work that the track addresses.

Level Displays the level in which the track is a level member. This field is blank until the level is committed, and remains blank if the track is committed without a level.

Owner Displays the user ID of the track owner.

Reference Displays an optionally assigned value that groups related defects and features.

Created Displays the date and time (yy/mm/dd hh:mm:ss) on which the track was created.

Updated Displays the date and time (yy/mm/dd hh:mm:ss) on which the track was last changed.

Abstract Displays a summary of the defect or feature that the track addresses.

Performing Queries with the Open Track List Dialog Box

The Open Track List dialog box allows you to perform queries and list information in the CMVC - Tracks window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to "Defining Default Queries and Tasks" on page 19.

To perform a query using the Open Track List dialog box:

1. Select `Open list...` from the File menu to display the Open Track List dialog box.
2. Enter the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

   Prefixes Prefixes specified when the defects or features were opened. Your family administrator can configure a list of choices for your development environment. You can also use the `Choices...` button to make your selection from a list of available prefixes.

   Defects/Features Identifiers of the features or defects that are being tracked.

   Releases Releases in which the work is being tracked. For example, you may want to list information on tracks associated with the `debugger2` release.

   States Particular states of the tracks. Valid states are `approve`, `fix`, `integrate`, `commit`, `test`, or `complete`. For example, you may want to list tracks that are in the `test` state.

   Targets Target, such as a level or a date, for performing the work that the track addresses. For example, you may want to list the track that will be applied towards level `014`, or those for which work is expected to be done by `93/12/22`.

   Levels Committed level in which the track was included. (CMVC updates a level when it is committed.) For example, you may want to list tracks that were level members in level `011` when that level was committed.

Owner IDs User IDs of the track owner. For example, you may want to list tracks that patjones owns.

References Optionally assigned value that groups related defects and features.

Add dates Dates (yy/mm/dd) on which the tracks were created. For example, you may want only tracks that were created after 93/12/29. Valid SQL operators for this field are <, >, like, is null, and between.

Change dates Dates (yy/mm/dd) on which the tracks were last changed. For example, you may want only tracks that were created after 93/12/14. Valid SQL operators for this field are <, >, like, is null, and between.

Abstracts Summary of the defect or feature that the track addresses. For example, you may want to list all of the tracks dealing with the defects or features having the keyword resource in them. You could do this by using the wildcard character (%) along with the SQL like operator and searching for the string %resource%.

3. Select OK to perform the query. The CMVC - Tracks window is refreshed, and the information that you requested is displayed in the client area.

You can use the Refresh now item on the File menu to reissue a query to the server. The most recent query is sent and the list is updated with current information from the server. If you issue a query to the server using the Open list... option, you can replace the query with the default query specified in the system file by selecting the Default list item from the File menu.

Note: For more information about how to use the Options menu items to specify operators, refer to “SQL Operator (in) and No Sort Combo Boxes” on page 11.

Performing Actions with the CMVC - Tracks Window

This section describes the authority and prerequisites for performing common actions with the CMVC - Tracks window, and how to perform these actions.

Creating a Track

Tracks are required for defects or features that apply to releases for which the tracking subprocess is included. A track is created automatically in defects or features for each sizing record that is in the accept state. Otherwise, a track has to be created manually. The track life cycle begins when the defect or feature is in the working state.

A track does not have a single name. Rather, a track is referred to by specifying a release name and a defect or feature name.

Note: You can also use the CMVC - Defects or CMVC - Features windows to create a track.

Authority Required: The defect owner has implicit authority. You have explicit authority if the TrackCreate action is specified for your user ID in the component associated with the defect.

Prerequisites: When creating tracks, you must ensure that the following requirements are met:

- The tracking subprocess is included in the process for the release
- The defect or feature is in the `working` or the `verify` state.

Creating tracks is a mandatory task for defects or features if the tracking subprocess is included in the process for the associated release. Creating a track leads to more effective change control.

Procedure: To create tracks:

1. Select `Create...` from the Actions menu to display the Create Tracks dialog box.
2. Type information in the following fields:

   Defects/Features Identifier of the defect or feature that you want to track.

   Releases Name of existing releases in which changes will be tracked to resolve the defects or implement the features. You can also import selected releases from the CMVC - Releases window.

   Target A reference, such as a level or a date, for the work being done for the tracks. You can type up to 15 alphanumeric characters. You can use all characters except vertical bars (|), ASCII control characters or shell metacharacters.

   Owner User ID of the track owner. If you do not specify an owner, then the default is the user creating the track. You can import a selected user ID from the CMVC - Users window.
3. Select `OK` to create the track.

One track is created for every specified defect or feature in every specified release. For example, four tracks are created if you specify defects `1155` and `1323`, and releases `code1` and `books1` in the Create Tracks dialog box. The following tracks are included:

- Two tracks that refer to fixes required in the release for source code
- Two tracks that refer to the same fixes required in the release for books documenting the application generated by that source code.

If there are any sizing records in the `accept` state when a track is created automatically, then one fix record is created for each of those sizing records. No fix records are created for sizing records in the `reject` or `abstain` states.

Because you created the track, you become the owner of the track associated with that release, unless a different user is specified.

While the defect or feature is in the `working` or `verify` state, you can create additional tracks if you find that the same defect applies to other releases. For example, defect `1159` was opened to correct a problem that applies to release4. When you examine the problem in more detail, you discover that older maintenance releases (`release2` and `release3`) also need this fix applied to them.

Note: As a defect owner or feature owner who created a track, you can cancel that track. If you reassign a track to another owner, the new track owner does not have implicit authority to cancel the track.

Canceling a Track

You can cancel a track if the defect or feature does not need to be addressed or implemented for the specified release.

Authority Required: The defect owner or feature owner has implicit authority. You have explicit authority if the TrackCancel action is specified for your user ID in the component associated with the defect or feature.

Prerequisites: When canceling a track, you must ensure that the following requirements are met:

- Track is not in the `commit`, `test`, or `complete` state
- Track is not a level member
- Files that reference the track were not changed.

To check that these criteria are met, refer to “Showing the Details of Tracks” on page 108.

Procedure: To cancel a track:

1. Select the track you want to cancel from the list of tracks in the client area.
2. Select `Cancel...` from the Actions menu to display the Cancel Tracks dialog box. Accept the prefilled information in the `Defects/Features` and `Releases` fields or type new information.
3. Select `OK` to cancel the track.

When you cancel a track, any association between the defect or feature and the release is also canceled. No further action is required.

Note: If you do not want to select tracks from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Moving a Track from a Fix State to an Integrate State

You can force a track from the `fix` state to the `integrate` state to indicate that no more fixes will be done for the track.

Normally, the track moves to the `integrate` state when all of the fix records for the track move to the `complete` state. When file changes are not required, however, you can force the track to the `integrate` state. For example, if a fix for another defect also fixes the current defect, then you do not have to make file changes for this track and you can force the track to the `integrate` state.

Authority Required: The track owner has implicit authority. You have explicit authority if the TrackIntegrate action is specified for your user ID in the component associated with the release.

Prerequisites: When moving tracks to the `integrate` state, you must ensure that the following requirements are met:

- Track is in the `fix` state
- Fix records are marked as completed.

Procedure: To move a track to the `integrate` state:

1. Select the track that you want to move from `fix` state to `integrate` state.
2. Select `Integrate...` from the Actions menu to display the Integrate Tracks dialog box. Accept the prefilled information in the `Defects/Features` and `Releases` fields, or type new information.
3. Select `OK` to move the track to the `integrate` state. You can now commit the track.

Note: If you do not want to select tracks from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Moving a Track to the Fix State

If you want to make additional changes to files that have associated tracks, you can force a track from the `integrate` state back to the `fix` state.

Authority Required: The track owner has implicit authority. You have explicit authority if the TrackFix action is specified for your user ID in the component associated with the release.

Prerequisites: When moving a track to the `fix` state, you must ensure that the following requirements are met:

- Track is in the `integrate` state.
- Track is not a level member.
- If the track is a level member, you must first remove the track from the level.

Procedure: To move the track to the `fix` state:

1. Select the track you want to move from `integrate` state to `fix` state from the list of tracks in the client area.
2. Select `Fix...` from the Actions menu to display the Fix Tracks dialog box. Accept the prefilled information in the `Defects/Features` and `Releases` fields, or type new information.
3. Select `OK` to move the track to the `fix` state. You must reactivate the appropriate fix record so that you can make changes to the files.

Note: If you do not want to select tracks from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Adding Corequisite Tracks

You can group tracks to indicate that work being done on a file that is associated with a specific defect or feature affects a file that is associated with another feature or defect. Adding tracks to form a corequisite group of tracks ensures that all interrelated files are included in a level.

Authority Required: The track owner has implicit authority. You have explicit authority if the CoreqCreate action is specified for your user ID in the component associated with the release.

Prerequisites: The tracks that you want to add must be in either the `fix` or `integrate` state.

Procedure: To add a corequisite track:

1. Select the track you want to identify as belonging to a corequisite group of two or more tracks from the list of tracks in the client area.
2. Select `Add...` from the Corequisites menu to display the Add Corequisite Tracks dialog box. Accept the prefilled information in the `Defects/Features` and `Release` fields or type new information.
3. Select `OK` to add the corequisite tracks.

You can add more tracks to the group simply by specifying one track from the existing corequisite group and specifying the new track or tracks to be added to the group.

You can merge a number of corequisite groups into a larger corequisite group simply by specifying one track from each of the corequisite groups you want to merge. To find out how to check whether a track is part of a corequisite group, see "Checking a Track for Dependencies" on page 105.

A level that includes one or more tracks from a corequisite group cannot be committed until all of the tracks in that corequisite group are included in the level.

Note: If you do not want to select tracks from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Removing Corequisite Tracks

You can remove a track from a corequisite group if the work being done on the file that is associated with a defect or feature does not affect the file that is associated with another defect or feature.

Authority Required: The track owner has implicit authority. You have explicit authority if the CoreqDelete action is specified for your user ID in the component associated with the release.

Prerequisites: The track must be in either the `fix` state or the `integrate` state.

Procedure: To remove a track from a corequisite group:

1. Select the track you want to remove from a corequisite group of two or more tracks from the list of tracks in the client area.
2. Select `Remove...` from the Corequisites menu to display the Remove Corequisite Tracks dialog box. Accept the prefilled information in the `Defects/Features` and `Release` fields or type new information.
3. Select `OK` to remove the track from the corequisite group.

Note: If you do not want to select tracks from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

The track is no longer associated with the other track or tracks. The other tracks must still be committed together in the same level. The track you just removed, however, does not have to be a level member in that level for that level to be committed.

The track will still have to be committed in a level eventually, but not necessarily in the same level as the rest of the tracks that have been identified as corequisites.

Checking a Track for Dependencies

You can check a track for dependencies to determine whether a track has any corequisite tracks or *prerequisite tracks*.

Authority Required: The track owner has implicit authority. You have explicit authority if the TrackCheck action is specified for your user ID in the component associated with the release.

Prerequisites: When checking a file for dependencies, you must ensure that the following requirements are met:

- Track is defined
- Track is part of a committed level.

Procedure: To check a track for dependencies:

1. Select the track you want to check for dependencies from the list of tracks in the client area.
2. Select `Check...` from the Actions menu to display the Check Tracks dialog box. Accept the prefilled information in the `Defects/Features` and `Releases` fields or type new information.
3. If required, type the name of a committed level in the `Level` field. This is the level that references the work being done for the tracks you want to check for dependencies.

   When you type information in this field, CMVC ignores dependencies existing outside of the specified level. You can also import a selected level from the CMVC - Levels window.
4. Select `OK` to check the dependencies.

Checks for corequisite tracks are based on previously defined corequisite relationships. For more information, see “Adding Corequisite Tracks” on page 103.

Checks for prerequisite tracks are based on previous changes to files associated with a given track. If you specify a committed level, prerequisites are only checked back as far as the version of the file made for that committed level. If you do not specify a level, all versions of the file are checked for prerequisites. Prerequisite tracks are maintained automatically in CMVC.

Note: If you do not want to select tracks from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Moving a Track from an Integrate State to a Commit State

Usually the track changes automatically from the `integrate` state to the `commit` state when a level is committed.

You can force the track to the `commit` state without committing the level if you do not want to make any changes based on the track. By doing this, you force the track through states so that you can close the defect or feature that the track addresses.

Authority Required: The track owner has implicit authority. You have explicit authority if the TrackCommit action is specified for your user ID in the component associated with the release.

Prerequisites: When moving tracks from the `integrate` state to the `commit` state, you must ensure that the following requirements are met:

- Track is in the `integrate` state
- Track does not have any file changes associated with it if the level subprocess is included in the process for the release.

Procedure: To move a track from the `integrate` state to the `commit` state:

1. Select the track you want to move from `integrate` state to `commit` state from the list of tracks in the client area.
2. Select `Commit...` from the Actions menu to display the Commit Tracks dialog box. Accept the prefilled information in the `Defects/Features` and `Releases` fields or type new information.
3. Select `OK` to move the track to the `commit` state.

Note: If you do not want to select tracks from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Moving a Track from a Commit State to a Test State

Usually, if the test subprocess is included in the process for a release, the track moves to the `test` state when a level moves to the `complete` state or when the track is committed without a level.

You can force the track to the `test` state without completing the level if you do not make any changes based on the track. This allows you to close the defect or feature that the track addresses.

Authority Required: The track owner has implicit authority. You have explicit authority if the TrackTest action is specified for your user ID in the component associated with the release.

Prerequisites: The track is in the `commit` state.

Procedure: To move a track from the `commit` state to the `test` state:

1. Select the track you want to move from `commit` state to `test` state from the list of tracks in the client area.
2. Select `Test...` from the Actions menu to display the Test Tracks dialog box. Accept the prefilled information in the `Defects/Features` and `Releases` fields or type new information.
3. Select `OK` to move the track to the `test` state. If there is no environment list, the track changes to the `complete` state.

If there is an environment list, the testers must test the changes in the specified environment and then mark the results on the test records. The track is in the `test` state until all test records are marked.

Note: If you do not want to select tracks from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Moving a Track from a Test State to a Complete State

If file changes are associated with a track, the track moves from the `test` state to the `complete` state when all test records are marked.

You can force a track from the `test` state to the `complete` state if there are no file changes associated with a track.

Authority Required: The track owner has implicit authority. You have explicit authority if the TrackComplete action is specified for your user ID in the component associated with the release.

Prerequisites: When moving tracks from the `test` state to the `complete` state, you must ensure that the following requirements are met:

- Track is in one of the `fix`, `integrate`, `commit` or `test` states.
- If the track is in the `fix` or `integrate` state, there must be no active changes associated with the track.

Procedure: To move a track from the `test` state to the `complete` state:

1. Select the track you want to move from the `test` state to the `complete` state from the list of tracks in the client area.
2. Select `Complete...` from the Actions menu to display the Complete Tracks dialog box. Accept the prefilled information in the `Defects/Features` and `Releases` fields or type new information.
3. Select `OK` to move the track to the `complete` state.

Once a track is in the `complete` state, you cannot change the state. This can cause:

- Defect or feature verification process to begin
- Defect or feature to move to the `closed` state if this is the last track and if all verification records are marked, or the verification subprocess is not included in the associated component process.

Note: If you do not want to select tracks from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Modifying the Ownership of a Track

You can assign a track to another user ID so that the owner of the user ID assumes the track owner responsibilities.

Authority Required: The track owner has implicit authority. You have explicit authority if the TrackAssign action is specified for your user ID in the component associated with the release.

Prerequisites: The track can be in any state.

Procedure: To modify a track owner:

1. Select the track for which you want to modify the owner from the list of tracks in the client area.
2. Select `Owner...` from the Modify menu to display the Modify Track Owner dialog box. Accept the prefilled information in the `Defects/Features` and `Releases` fields or type new information.

3. Type the user ID of the new owner of the specified track in the New owner field. You can also import a selected user ID from the CMVC - Users window.
4. Select `OK` to modify the owner.

Note: If you do not want to select tracks from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Modifying the Target for a Track

You can modify the target for the track.

Authority Required: The track owner has implicit authority. You have explicit authority if the TrackModify action is specified for your user ID in the component associated with the release.

Prerequisites: The track can be in any state.

Procedure: To modify the track target:

1. Select the track for which you want to modify properties from the list of tracks in the client area.
2. Select `Properties...` from the Modify menu to display the Modify Track Properties dialog box. Accept the prefilled information in the `Defects/Features` and `Releases` fields or type new information.
3. Type a target, such as a level or a date, for the work being done for the tracks in the `Target` field. You can type up to 15 alphanumeric characters. For example, the work on the tracks will be completed by `94/03/25` instead of `93/12/31`, as originally thought. You can use all characters except vertical bars (`|`), ASCII control characters or shell metacharacters.
4. Select `OK` to modify the target.

Note: If you do not want to select tracks from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Showing the Details of Tracks

You can show the following details of tracks:

- Track approval records
- Fix records and test records for the track
- Levels in which the track is a member
- File changes associated with the track.

Authority Required: The track owner has implicit authority. You have explicit authority if the TrackView action is specified for your user ID in the component associated with the release.

Prerequisites: The track can be in any state.

Procedure: To show information about more than one track:

1. Select all of the tracks about which you want to show information from the list of tracks in the client area.
2. Select `Details` from the Show menu. The CMVC - Information window is opened and the track details are displayed in the client area.

Note: If you do not want to select tracks from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Showing the Note History for a Track

You can show the notes that are related to a specific track.

Authority Required: None.

Prerequisites: When showing the note history for a track, you must ensure that the following requirements are met:

- The track is defined
- You have access to the family to which the track belongs.

Procedure: To show the note history for more than one track:

1. Select the track for which you want to display the note history from the list of tracks in the client area.
2. Select `Note history` from the Show menu. The CMVC - Information window is opened and the information that you requested is displayed in the client area.

Showing the Change History of a Track

You can show the history of changes for a track or tracks by selecting the Change history option from the Show menu. For more information about how to show the change history, refer to “Using the CMVC - Change History Window” on page 23.

Additional Show Options

Additional Show options allow you to open certain GUI windows, including the CMVC - Approval Records, CMVC - Fix Records, CMVC - Test Records, and CMVC - Level Members windows. To open these windows, select the corresponding option from the Show menu.

Chapter 8. Working with Levels and Level Members

You can use the CMVC - Levels and CMVC - Level Members windows to perform a variety of actions on levels and level members. This chapter describes the actions that you can perform as a level owner. If you have superuser authority, you can perform any of the actions described in this chapter.

Note: The actions that are described apply only if the tracking or level subprocess is included in the process for the release.

Level Owner’s Tasks

A release owner can create a level to represent a collection of file changes made in a release. By default, the release owner becomes the level owner.

Some of the actions that you can perform with the CMVC - Levels window as a level owner include:

- Create a level
- Check a level for dependencies
- Extract the files of a level to create a file tree
- Commit and complete a level
- Delete a level
- Modify the name, ownership, or type of a level
- Add a level member
- Show the details or change history of a level.

Some of the actions that you can perform with the CMVC - Level Members window as a level owner include:

- Adding a track to a level
- Removing a track from a level.

States of a Level

The state of a level indicates the progress that is being made in addressing the collection of changes in a release. You create a level so that a collection of file changes can be easily referenced.

When you create a level, that level is in the `working` state. In the `working` state a level is not associated with any tracks.

During the `working` or `integrate` state of a level, tracks can be designated as level members. Tracks must be in the `fix` or `integrate` state before they can become level members. All level members, however, must be in the `integrate` state before a level can be committed.

A level moves from the `working` to the `integrate` state when the first level member is added. Additional level members can be added while the level is in the `integrate` state.

Once a level is committed, all file changes that are associated with its level members become permanent and are reflected in the release. File changes that are associated with that level are committed and cannot be modified or deleted after that time.

When you complete a level, you indicate that a level is ready for testing. This activates any test records that exist for the track. If no test records exist, the track moves to the `complete` state.

Displaying Level Information in the Client Area

The CMVC - Levels window allows you to work with levels in CMVC. When you query a level, information is displayed in the client area of the window under the following headings:

Level Displays the name of the level.

Release Displays the existing release in which the file changes apply.

State Displays the current state of a level. Valid states are `working`, `integrate`, `commit`, and `complete`.

Level Type Displays the type of level. Your family administrator can configure these values and establish a default type. The shipped level types are `development`, `production`, `integration`, `prototype` and `other`.

Owner Displays the user ID of the level owner.

Area Displays the area, such as, the department, of the level owner.

Created Displays the date and time (yy/mm/dd hh:mm:ss) on which the level was created.

Committed Displays the date and time (yy/mm/dd hh:mm:ss) on which the level was committed.

Updated Displays the date and time (yy/mm/dd hh:mm:ss) on which the level was last updated.

Performing Queries with the Open Level List

The Open Level List dialog box allows you to perform queries and list information in the CMVC - Levels window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to “Defining Default Queries and Tasks” on page 19.

To perform a query with the Open Level List dialog box:

1. Select `Open list...` from the File menu to display the Open Level List dialog box.
2. Enter the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

   Levels Name of the level.

   Releases Releases in which the file changes apply. For example, you may want to list information on the levels that are associated with the `debugger2` release.

   States The particular states of the levels. For example, you may want to list levels that are in the `integrate` state.

Types Type of level. Your family administrator can configure a list of choices. Use the `Choices...` button to make your selection from a list of available alternatives.

Owner IDs User ID of the level owner. For example, you may want to list the levels that `jroberts` owns.

Owner areas Department or area in which the level owner works. For example, you may want to list the levels that are owned by the users in the area `tools`.

Add dates Date (yy/mm/dd) on which the level was created. For example, you may want to list levels that were created after `93/04/29`. Valid SQL operators for this field are <, >, `like`, `null`, and `between`.

Commit dates Date (yy/mm/dd) on which the level was committed. For example, you may want to list levels that were committed after `93/04/29`. Valid SQL operators for this field are <, >, `like`, `null`, and `between`.

Change dates Date (yy/mm/dd) on which the level was updated. For example, you may want to list levels that were updated after `93/04/29`. Valid SQL operators for this field are <, >, `like`, `null`, and `between`.

3. Select `OK` to perform the query. The CMVC - Levels window is refreshed, and the information that you requested is displayed in the client area.

You can use the `Refresh now` item on the File menu to reissue a query to the server. The most recent query is sent and the list is updated with current information from the server. If you issue a query to the server using the `Open list...` option, you can replace the query with the default query specified in the system file by selecting the `Default list` item from the File menu.

Note: For more information about how to use the Options menu items to specify operators, refer to “SQL Operator (in) and No Sort Combo Boxes” on page 11.

Performing Actions with the CMVC - Levels Window

This section describes the authority and prerequisites for performing actions with the CMVC - Levels window and how to perform these actions.

Creating a Level

You can create levels to group file changes within the release.

Authority Required: The release owner has implicit authority. You have explicit authority if the LevelCreate action is specified for your user ID in the component associated with the release.

Prerequisites: When creating a level, you must ensure that the following requirements are met:

- Release exists in CMVC
- The release process includes the tracking and level subprocesses

Procedure: To create a level:

1. Select `Create...` from the Actions menu to display the Create Levels dialog box.
2. Type information in the following fields:

   Levels Unique name that refers to the collection of file changes. You can type up to 15 alphanumeric characters. You can use all characters except vertical bars (|), ASCII control characters or shell metacharacters.

   Release Name of an existing release that contains the collection of file changes that are grouped together by the level. You can also import a selected release from the CMVC - Releases window.

   Type Type of level that is being created. Your family administrator can configure a list of choices. You can use the `Choices...` button to make your selection from a list of available alternatives.
3. Select `OK` to create the level. You become the level owner by default.

Extracting a File Tree for a Level

You can extract a full file tree or a *delta file tree*.

Authority Required: The level owner has implicit authority. You have explicit authority if the LevelExtract action is specified for your user ID in the component associated with the release.

Prerequisites: When extracting a file tree for a level, you must prepare the directory that is going to receive the extracted level. You must:

- Create the directory if it does not already exist.
- Export the directory so the server can mount it and write the files into it.

Procedure: To extract a level:

1. Select the level you want to extract from the list of levels in the client area.
2. Select `Extract...` from the Actions menu to display the Extract Level dialog box. Accept the prefilled information in the `Levels` and `Release` fields, or type new information.
3. Specify the type of file tree you want to extract. The types include:

   Delta When you extract a delta file tree, you create a directory structure that contains only the files that were changed for this level. You can extract a delta file tree for either committed or not committed levels. This is the default tree type.

   Full When you extract a full file tree, you create a directory structure that contains all files, regardless of whether they were changed or not, for this level. You can extract full file trees only if the level has been committed.
4. Type the destination host and directory information in the following fields:

   Host Host on which the extracted files are to be placed.

   Directory Directory on the host in which the extracted files are to be placed. This directory must already be exported via the *Network File System** (NFS**)*.

5. Type information for the extracted files in the following fields:

   User number (UID) Internal number that uniquely identifies the user to the system. All extracted files are owned by this user. Examples are 200 or 104. The default is the CMVC family's *User Number (UID)*.

   Group number (GID) Internal number that uniquely identifies the group to the system. All extracted files are owned by this group. Examples are 1 or 4. The default is the CMVC family's *Group Number (GID)*.

   File permissions Read, write, and execute file permissions in octal notation. The default is the file mode associated with the file stored on the CMVC server, with read-only access for all users.

   Directory permissions Read, write, and execute directory permissions in octal notation. The default 750 represents read, write, and execute access for the owner, read and execute access for users in the owner's group, and no access for all other users.

6. If you do not want to substitute assigned values for keywords imbedded in the extracted files, select Expand keywords. The default is to substitute values for the keywords.

7. If your target host is running the OS/2 operating system and you have indicated the target directory using the proper OS/2 file-naming convention, CMVC adds the Carriage Return and Line Feed characters to all extracted files as a default.

   If your target host is running AIX, SunOS, or HP-UX, click on the CRLF button to add Carriage Return and Line Feed characters to the extracted files.

8. Select OK to extract the level.

The contents of any files in that level when it was committed are accessed. When you destroy a file, the database record for the file is destroyed but the contents of the file remain on the server, so previously committed levels can be re-built.

The CMVC server unmounts the destination directory once the level extract is complete.

To remove files which have been deleted and renamed from an extracted file tree, CMVC creates a file named **CMVC.GON** that specifies the full path name of each deleted or renamed file. This file is extracted with the files in a delta tree extraction. Extracting the delta file tree of a level which has not been committed will extract all current

If you do not want to select the level from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Checking a Level for Dependencies

Checking a level helps you to determine whether or not a level is ready to be committed. You cannot commit a level until all of its prerequisite and corequisite tracks are included in the level.

Authority Required: The level owner has implicit authority. You have explicit authority if the LevelCheck action is specified for your user ID in the component associated with the release.

Prerequisites: The level is in the `integrate` state.

Procedure: To check a level for dependencies:

1. Select the level that you want to check for outstanding prerequisite tracks and outstanding corequisite tracks from the list of levels in the client area.
2. Select `Check...` from the Actions menu to display the Check Level dialog box. Accept the prefilled information in the `Levels` and `Release` fields or type new information.
3. Select one of the following radio buttons to indicate how much information about the dependencies you require:

   Tracks only Displays the defect or feature numbers referencing any corequisite or prerequisite tracks that are not included in the level. This is the default.

   Tracks and files Displays the defect or feature numbers of the corequisite or prerequisite tracks that are not included in the level, as well as the track of which they are corequisites or prerequisites, and the files associated with each of those tracks.
4. Select `OK` to check the level for dependencies. A CMVC - Information window is opened. It contains a list of any remaining dependencies (that is, corequisite or prerequisite tracks) which should be added to the level in order to eventually commit the level.

Note: If you do not want to select the level from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Committing a Level

You can commit a level when all of its level member tracks are in the `integrate` state. You cannot commit a level if one or more tracks which are level members have prerequisite or corequisite tracks that are not level members of the level.

Authority Required: The level owner has implicit authority. You have explicit authority if the LevelCommit action is specified for your user ID in the component associated with the release.

Prerequisites: All the level members are in the `integrate` state.

Procedure: To commit a level:

1. Select the level that you want to commit from the list of levels in the client area.
2. Select `Commit...` from the Actions menu to display the Commit Levels dialog box. Accept the prefilled information in the `Levels` and `Release` fields or type new information.
3. Select `OK` to commit the level.

Once a level is committed, it cannot be modified or deleted.

Note: If you do not want to select the level from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Completing a Level

You can move a level to the `complete` state so as to indicate it is ready for testing.

Authority Required: The level owner has implicit authority. You have explicit authority if the LevelComplete action is specified for your user ID in the component associated with the release.

Prerequisites: None.

Procedure: To complete a level:

1. Select the level that you want to complete from the list of levels in the client area.
2. Select `Complete...` from the Actions menu to display the Complete Levels dialog box. Accept the prefilled information in the `Levels` and `Release` fields or type new information.
3. Select `OK` to complete the level.

The level is moved to the `complete` state. Any test records that exist for the track are activated. If no test records exist, the track moves to the `complete` state.

Note: If you do not want to select the level from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Deleting a Level

You can delete a level if a level is not required or if no changes were integrated to the level.

Authority Required: The level owner has implicit authority. You have explicit authority if the LevelDelete action is specified for your user ID in the component associated with the release.

Prerequisites: The level is in the `working` or `integrate` state.

Procedure: To delete a level:

1. Select the level that you want to delete from the list of levels in the client area.
2. Select `Delete...` from the Actions menu to display the Delete Level dialog box. Accept the prefilled information in the `Levels` and `Release` fields or type new information.
3. Select `OK` to delete the level.

Note: If you do not want to select the level from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Adding a Level Member to a Level

You can add a level member to an existing level.

Authority Required: The level owner has implicit authority. You have explicit authority if the MemberCreate action is specified for your user ID in the component associated with the release.

Prerequisites: When adding a level member to a level, you must ensure that the following requirements are met:

- Level is not in the `commit` or `complete` state
- Track is in the `fix` or `integrate` state.

Procedure: To add a level member to a level:

1. Select `Add level members...` from the Actions menu to display the Add Level Members dialog box.
2. Type information in the following fields:

   Level Level to which you want to add the level member.

   Release Name of the release that contains the level to which you want to add the level members. You can also import a selected release from the CMVC - Releases window.

   Tracks Identifier of the tracks that you want to add as level members. You can also import selected tracks from the CMVC - Tracks window.
3. Select `OK` to add the track to the level.

A single track can be a member of more than one level. The file changes associated with the track, however, are committed when the first level to which the track belongs is committed.

Note: You can also add level members with the CMVC - Level Members window. For more information, see “Adding a Track to a Level” on page 122.

Modifying the Name of a Level

You can rename a level.

Authority Required: The level owner has implicit authority. You have explicit authority if the LevelModify action is specified for your user ID in the component associated with the release.

Prerequisites: The level is in the `working` or `integrate` state

Procedure: To rename a level:

1. Select the level that you want to rename from the list of levels in the client area.
2. Select `Name...` from the Modify menu to display the Modify Level Name dialog box. Accept the prefilled information in the `Level` and `Release` fields or type new information.
3. Enter a new name for the level in the `New name` field. You can type up to 15 alphanumeric characters. You can use all characters except vertical bars (`|`), ASCII control characters or shell metacharacters.
4. Select `OK` to rename the level. All members of that level are modified to reflect the new level name.

Note:

If you do not want to select the level from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Reassigning the Ownership of a Level

You can transfer ownership of a level to another user.

Authority Required: The level owner has implicit authority. You have explicit authority if the LevelAssign action is specified for your user ID in the component associated with the release.

Prerequisites: The level can exist in any state.

Procedure: To transfer ownership of a level to another user:

1. Select the level you want to reassign to another user from the list of levels in the client area.
2. Select `Owner...` from the Modify menu to display the Modify Level Owner dialog box. Accept the prefilled information in the `Levels` and `Release` fields or type new information.
3. Type the user ID of the new owner in the `New owner` field. You can also import a user ID from the CMVC - Users window.
4. Select `OK` to modify the owner of the level. The new owner assumes ownership responsibilities for the level.

Note: If you do not want to select the level from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Modifying the Properties of a Level

You can modify the type of level.

Authority Required: The level owner has implicit authority. You have explicit authority if the LevelModify action is specified for your user ID in the component associated with the release.

Prerequisites: The level can exist in any state.

Procedure: To modify the level type:

1. Select the level for which you want to modify properties from the list of levels in the client area.
2. Select `Properties...` from the Modify menu to display the Modify Level Properties dialog box. Accept the prefilled information in the `Levels` and `Release` fields or type new information.
3. Specify the type of level in the `Type` field. Your family administrator can configure a list of choices. You can use the `Choices...` button to make your selection from a list of available alternatives.
4. Select `OK` to modify the level type.

Note: If you do not want to select the level from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Showing the Details of a Level

You can show the details of a level, including:

- Owner's user ID, name, and area
- Date on which the level was created, committed, and last updated
- State of the level
- Existing level member tracks and the abstract for the defects or features that the tracks address.

Authority Required: The level owner has implicit authority. You have explicit authority if the LevelView action is specified for your user ID in the component associated with the release.

Prerequisites: The level is in any state.

Procedure: To show details for a single level, double-click on a level listed in the client area.

Another way to show the details of a level is:

1. Select the level you want to display from the list of levels in the client area.
2. Select `Details` from the Show menu. The CMVC - Information window is opened and the information that you requested is displayed.

Note: If you do not want to select the level from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Showing the Change History of a Level

You can show the history of changes of a level.

Authority Required: None.

Prerequisites: When showing the change history for a level, you must ensure that the following requirements are met:

- Level exists in CMVC.
- You must have a valid CMVC user ID for the family.

Procedure: To show the change history for a level:

1. Select the level for which you want to display the change history from the list of levels in the client area.
2. Select `Change history` from the Show menu. The CMVC - Change History window is opened and the information that you requested is displayed in the client area.

Note: For more information about the CMVC - Change History window, refer to "Using the CMVC - Change History Window" on page 23.

Displaying Level Member Information in the Client Area

The CMVC - Level Members window allows you to specify the tracks that you want to include in a level. A track can become a level member when the track is in the `fix` or `integrate` state. A track represents a set of changed files.

When you query a level member, information is displayed in the client area of the window under the following headings:

Level Displays the name that is assigned to the level.

Releases Displays the name of the existing release in which the file changes apply.

Prefix Displays the value specified when the defect or feature is opened. Your family administrator can configure these values.

Number Displays the alphanumeric identifier for the defect or feature.

Reference Displays an optionally assigned value that groups related defects or features.

Track Owner Displays the user ID of the track owner.

Track Owner's Name Displays the name of the track owner.

Area Displays the area, such as a department, in which the track owner works.

Performing Queries with the Open Level Members List Dialog Box

The Open Level Members List dialog box allows you to perform queries and list information in the CMVC - Level Members window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to "Defining Default Queries and Tasks" on page 19.

To perform a query using the Open Level Members List dialog box:

1. Select `Open list...` from the File menu to display the Open Level Member List dialog box.
2. Enter the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

   Levels Name of the levels. For example, you may want to list level member information for specific levels.

   Releases Name of the release in which the file changes apply. For example, you may want to list level member information on the levels that are associated `debugger2` release.

   Prefixes Values that are specified for the defects or features. Your family administrator can configure a list of choices. You can use the `Choices...` button to make your selection from a list of available alternatives.

Tracks Alphanumeric identifiers for the defects or features that are being tracked. For example, you may want to list level information for specific tracks.

References Optionally assigned values for defects or features that are used to group related defects and features.

Track owner IDs User ID of the track owner. For example, you may want to list level member information for tracks owned by `jroberts`.

Track owner names Name of the track owner. For example, you may want to list level member information for tracks owned by `Percy G. Williams`, but you do not know Percy's user ID.

Track owner areas Area of track owner. For example, you may want to list level information for tracks that are owned by the `production` department.

3. Select `OK` to perform the query. The CMVC - Level Members window is refreshed, and the information that you requested is displayed in the client area.

You can use the `Refresh now` item on the File menu to reissue a query to the server. The most recent query is sent and the list is updated with current information from the server. If you issue a query to the server using the `Open list...` option, you can replace the query with the default query specified in the system file by selecting the `Default list` item from the File menu.

Note: For more information about how to use the Options menu items to specify operators, refer to "SQL Operator (in) and No Sort Combo Boxes" on page 11.

Performing Actions with the CMVC - Level Members Window

This section describes the authority and prerequisites for performing actions with the CMVC - Level Members window and how to perform these actions.

Adding a Track to a Level

You can add a track to a level so that file changes associated with the track become permanent when the level is committed. If the level subprocess has been included in the associated release process, you must add all corequisite and prerequisite tracks to commit the level.

Authority Required: The level owner has implicit authority. You have explicit authority if the MemberCreate action is specified for your user ID in the component associated with the release.

Prerequisites: When adding a track to a level, you must ensure that the following requirements are met:

- Track is in the `fix` or `integrate` state
- Level is not in the `commit` or `complete` state.

Procedure: To add a track to a level:

1. Select `Add...` from the Actions menu to display the Add Level Members dialog box.

2. Type information in the following fields:

   Level Level to which you want to add a track as a level member. You can also import a selected level from the CMVC - Levels window.

   Release Name of the release that contains the level to which you want to add a level member. You can also import a selected release from the CMVC - Releases window.

   Tracks Alphanumeric identifier for the defect or feature that is being tracked by the track you want to identify as a level member. You can also import a selected track from the CMVC - Tracks window.

3. Select `OK` to add the track to the level. The track becomes a member of that level. It is referred to as a level member.

If the level was in the `working` state when a track was added, the level moves to the `integrate` state. If it was in the `integrate` state, it does not change states.

A single track can be a member of more than one level. The file changes associated with the track, however, will get committed when the first level gets committed.

You can now extract the level (delta extract if the level is not committed; full extract if the level is committed). For more information on extracting a file tree for a level, see "Extracting a File Tree for a Level" on page 114.

Removing a Track from a Level

You can remove a track from a level.

Authority Required: The level owner has implicit authority. You have explicit authority if the MemberDelete action is specified for your user ID in the component associated with the release.

Prerequisites: When removing a track from a level, you must ensure that the following requirements are met:

- Level is in the `integrate` state
- Track you want to remove is a member of that level.

Procedure: To remove a track from a level:

1. Select the level member you want to remove from a level from the list of level members in the client area.
2. Select `Remove...` from the Actions menu. A Remove Level Members dialog box is displayed. Accept the prefilled information in the `Level`, `Release`, and `Tracks` fields or type in new information.
3. Select `OK` to remove the level members.

If you remove all level members from a level, the level moves back to the `working` state, otherwise, the level stays in the `integrate` state. You can:

- Extract the level as long as it still contains level members
- Perform a delta extraction on a non-committed level
- Perform a full extraction on a committed level
- Commit the level even if it has no level members
- Check the level for prerequisites and corequisites

- Delete the level if all level members have been removed.

Note: If you do not want to select the level from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Chapter 9. Working with Approval, Fix, and Test Records

You can use the CMVC - Approval Records, CMVC - Fix Records, and CMVC - Test Records windows to perform a variety of actions on approval, fix, and test records. This chapter describes the role that approval, fix, and *test records* play in problem tracking and the actions that you can perform as a record owner. If you have superuser authority, you can perform any of the actions described in this chapter. Approval, Fix, and Test records are optional depending on whether their subprocesses are specified for a release process.

Approval Record Owner's Tasks

If the approval subprocess is included in the process for a release, *approval records* are created. Every time a new track is created for a release, approval records are created for each *approver* identified on the release *approver list*. For more information about release approver lists, refer to Chapter 12, "Working with Releases" on page 173.

Some of the actions that you can perform as an owner of approval records include:

- Create an approval record
- Accept or reject the changes proposed for a specific track
- Abstain from approving the changes proposed for a specific track
- Delete an approval record
- Show the note history of an approval record.

You can review the information recorded in the defect or feature and determine whether work can proceed on a track. You can also reassign an approval record so that another person assumes the role of approver.

Displaying Approval Record Information in the Client Area

The CMVC - Approval Records window allows you to work with approval records. When you query an approval record, specific information is displayed in the client area of the window under the following headings:

Prefix Displays the identifier specified when the defect or feature is opened.

Number Displays the identifier of the defect or feature that the track addresses.

Release Displays the release in which work to address the particular defect or feature is performed.

Owner Displays the user ID of the owner of the approval record. The owner is responsible for approving the proposed changes for the tracks.

State Displays the current state of the approval record. Valid states are `ready`, `accept`, `reject`, and `abstain`.

Reference Displays an optionally assigned value that groups related defects or features.

Created Displays the date and time (yy/mm/dd hh:mm:ss) on which the approval record was created.

Updated Displays the date and time (yy/mm/dd hh:mm:.ss) on which the approval record was last updated.

Abstract Displays a summary of the defect or feature.

Performing Queries with the Approval Record List Dialog Box

The Open Approval Record List dialog box allows you to perform queries and list information in the CMVC - Approval Records window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to "Defining Default Queries and Tasks" on page 19.

To perform a query using the Open Approval Record List dialog box:

1. Select `Open list...` from the File menu to display the Open Approval Record List dialog box.
2. Type the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

   Prefixes Identifier that describes the type of defect or feature that was opened. Your family administrator can configure these values.

   Defects/Features Identifier of the feature or defect to which the approval records refer. For example, you may want to list only the approval records for the tracks whose work addresses specific defects and features.

   Releases Release in which work is being tracked and which requires approval records to be completed before continuing work. For example, you may want to list approval records for tracks that are associated with the `debugger2` release.

   Owner IDs User IDs of the owners of the approval records. For example, you may want to list the approval records that `patjonesall` owns.

   States Particular state or states of approval records. Valid options are `ready`, `accept`, `reject`, and `abstain`. For example, you may want to list approval records that are in the `ready` state.

   References Displays an optionally assigned value that groups related defects and features.

   Add dates Date (yy/mm/dd) on which the defect or feature was created. For example, you may want to list only the approval records that were created after `93/12/29`. Valid SQL operators for this field are <, >, `like`, `is null`, and `between`.

   Change dates Dates (yy/mm/dd) on which the defect or feature was last changed. For example, you may want to list only the approval records that were changed prior to `93/12/13`. Valid SQL operators for this field are <, >, `like`, `is null`, and `between`.

Abstracts Summary of the defect or feature that the approval record addresses. For example, you may want to list all the approval records for defects or features having the keyword `resources` in them. You could do this by using the wildcard character (`%`) along with the SQL `like` operator and searching for the string `%resources`.

3. Select `OK` to perform the query. The CMVC - Approval Records window is refreshed, and the information that you requested is displayed in the client area.

You can use the `Refresh now` item on the File menu to reissue a query to the server. The most recent query is sent and the list is updated with current information from the server. If you issue a query to the server using the `Open list...` option, you can replace the query with the default query specified in the system file by selecting the `Default list` item from the File menu.

Note: For more information about how to use the Options menu items to specify operators, refer to “SQL Operator (in) and No Sort Combo Boxes” on page 11.

Performing Actions with the CMVC - Approval Records Window

This section describes the authority and prerequisites for performing common actions with the CMVC - Approval Records window, and how to perform these actions.

Creating an Approval Record

You can create approval records for the users who are not members of the associated release approver list.

Authority Required: The track owner has implicit authority. You have explicit authority if the ApprovalCreate action is specified for your user ID in the component associated with the release.

Prerequisites: The track is in the `approve` state.

Procedure: To create an approval record:

1. Select `Create...` from the Actions menu to display the Create Approval Records dialog box.
2. Type information in the following fields:

   Defects/Features Defect or feature identifier for the tracks for which you want to create an approval record. You can also import selected defects or features from the CMVC - Defects window or CMVC - Features window.

   Releases Names of the releases that are associated with the tracks for which you want to create an approval record. You can also import selected releases from the CMVC - Releases window.

   Approver User ID of the owner of the new approval record. You can also import selected users from the CMVC - Users window.
3. Select `OK` to create the approval records. Creating records does not change the approver list that is associated with the release. It only identifies these

users as additional approvers for the specific track. The approver list is not modified, but additional approval records are created and associated with the track.

Note: If you do not want to select tracks from a list in the client area, you can begin with step 2 and type the information directly in the text entry fields.

Accepting an Approval Record

When you accept an approval record, you allow work to proceed on the track.

Authority Required: The approval record owner has implicit authority. You have explicit authority if the ApprovalAccept action is specified for your user ID in the component associated with the release.

Prerequisites: When accepting approval records, you must ensure that the following requirements are met:

- Track is in the `approve` state
- Approval record can be in any state.

Procedure: To accept an approval record:

1. Select the approval record you want to accept from the list of records in the client area.
2. Select `Accept...` from the Actions menu. An Accept Approval Records dialog box is displayed. Accept the prefilled information in the `Defects/Features`, `Releases`, and `Approver` fields or type in new information.
3. Select `OK` to accept the approval record.

When all approval records are in either the `accept` or `abstain` state, the track moves to the `fix` state.

Note: If you do not want to select the approval records from a list in the client area, you can begin with step 2 and type the information directly into the text entry fields.

Abstaining from Approving an Approval Record

If you do not want to accept or reject the changes for a given track, but you want work to continue on the track, you must abstain from approving the approval record.

Authority Required: The approval record owner has implicit authority. You have explicit authority if the ApprovalAbstain action is specified for your user ID in the component associated with the release.

Prerequisites: When abstaining from approving a record, you must ensure that the following requirements are met:

- Track is in the `approve` state
- Approval record can be in any state.

Procedure: To abstain from approving a record:

1. Select the approval record for which you want to abstain from approval from the list of records in the client area.

2. Select Abstain... from the Actions menu. An Abstain From Approval dialog box is displayed. Accept the prefilled information in the Defects/Features, Releases, and Approver fields or type in new information.
3. Select OK to abstain from approving the record.

When all approval records are either in the accept or abstain state, the track moves to the fix state.

Note: If you do not want to select the approval records from a list in the client area, you can begin with step 2 and type the information directly into the text entry fields.

Deleting an Approval Record

You can delete an approval record when it is no longer required.

Authority Required: No one has implicit authority. You have explicit authority if the ApprovalDelete action is specified for your user ID in the component associated with the release.

Prerequisites: When deleting an approval record, you must ensure that the following requirements are met:

- Track is in the approve state
- Approval record can be in any state.

Procedure: To delete an approval record:

1. Select the approval record that you want to delete from the list of records in the client area.
2. Select Delete... from the Actions menu. A Delete Approval Records dialog box is displayed. Accept the prefilled information in the Defects/Features, Releases, and Approver fields or type in new information.
3. Select OK to delete the approval record.

Note: If you do not want to select the approval records from a list in the client area, you can begin with step 2 and type the information directly into the text entry fields.

Rejecting an Approval Record

If you do not want work to proceed on the track, then you must reject the approval record.

Authority Required: The approval record owner has implicit authority. You have explicit authority if the ApprovalReject action is specified for your user ID in the component associated with the release.

Prerequisites: When rejecting an approval record, you must ensure that the following requirements are met:

- Track is in the approve state
- Approval record can be in any state.

Procedure: To reject an approval record, use the following method:

1. Select the approval record you want to reject from the list of records in the client area.

2. Select `Reject...` from the Actions menu. A Delete Approval Records dialog box is displayed. Accept the prefilled information in the `Defects/Features`, `Releases`, and `Approver` fields or type in new information.
3. Select `OK` to delete the approval record.

The track moves to the `fix` state when all approval records are in the `accept` or `abstain` state only.

Approvers can change the state of their approval records to abstain or accept, after they have rejected the approval records.

Note: If you do not want to select the approval records from a list in the client area, you can begin with step 2 and type the information directly into the text entry fields.

Modifying the Ownership of an Approval Record

You can assign an approval record to another CMVC user ID to transfer the approver role.

Authority Required: The approval record owner has implicit authority. You have explicit authority if the ApprovalAssign action is specified for your user ID in the component associated with the release.

Prerequisites: When modifying the ownership of an approval record, you must ensure that the following requirements are met:

- Track is in the `approve` state
- Approval record can be in any state.

Procedure: To modify the ownership of an approval record:

1. Select the approval record for which you want to modify the owner from the list of records in the client area.
2. Select `Owner...` from the Modify menu. A Modify Approval Record Owner dialog box is displayed. Accept the prefilled information in the `Defects/Features`, `Releases`, and `Approver` fields or type in new information.
3. Type the user ID of the person who will be the new owner of the specified approval records in the `New approver` field. You can import a selected user ID from the CMVC - Users window.
4. Select `OK` to modify the owner. If the approval record was previously marked, it will be reset to the `ready` state. The new approval record owner must mark the approval record with `accept`, `reject`, or `abstain`.

The approver list associated with the release does not change if you reassign the approval record. For information about changing approver lists, refer to Chapter 12, “Working with Releases” on page 173.

Note: If you do not want to select the approval records from a list in the client area, you can begin with step 2 and type the information directly into the text entry fields.

Showing the Note History of an Approval Record

You can show the notes that are associated with a specific approval record.

Authority Required: None.

Prerequisites: When showing the note history for an approval record, you must ensure that the following requirements are met:

- The approval record exists in CMVC.
- You must have a valid CMVC user ID for the family for which the record is defined.

Procedure: To show the note history for a single approval record, double-click on an approval record listed in the client area.

To show the note history for one or more approval records:

1. Select the approval record for which you want to display the note history from the list of records in the client area.
2. Select `Note history` from the Show menu. The CMVC - Information window is opened and the information that you requested is displayed in the client area.

Showing the Change History of an Approval Record

You can show the history of changes for one or more approval records. For more information about how to show the change history, refer to “Using the CMVC - Change History Window” on page 23.

Fix Record Owner's Tasks

A fix record owner reviews all necessary file changes within a specific component and determines whether the changes are complete and ready to be integrated into a level or committed with a track. In this way, the fix record owner controls the transition of a track from the `fix` state to the `integrate`, `commit`, or `complete` states.

The component owner is usually the fix record owner, but ownership can be reassigned. As a fix record owner, you can perform the following actions:

- Create, complete, activate, or delete a fix record
- Modify the ownership of a fix record.

Displaying Fix Record Information in the Client Area

The CMVC - Fix Records window allows you to work with fix records. When you query a fix record, specific information is displayed in the client area of the window under the following headings:

Prefix Displays the identifier for a defect or feature that was specified when the defect or feature was opened.

Number Displays the identifier for the defect or feature that the fix record addresses.

Release Displays the release in which the work is being done to address the specified defect or feature.

Component Displays the name of the component that is associated with the file in which changes are needed for the track.

State Displays the current state of the fix record. Valid states are `notReady`, `ready`, `active`, or `complete`.

Owner Displays the user ID of the fix record owner.

Reference Displays an optionally assigned value that groups related defects and features.

Created Displays the date and time (yy/mm/dd hh:mm:ss) on which the fix record was created.

Updated Displays the date and time (yy/mm/dd hh:mm:ss) on which the fix record was last changed.

Abstract Displays a summary of the defect or feature.

Performing Queries with the Open Fix Record List Dialog Box

The Open Fix Record List dialog box allows you to perform queries and list information in the CMVC - Fix Records window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to "Defining Default Queries and Tasks" on page 19.

To perform a query using the Open Fix Record List dialog box:

1. Select `Open list...` from the File menu to display the Open Fix Record List dialog box.
2. Enter the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

   Prefixes Values that indicate whether the tracks reference a defect or feature. Your family administrator can configure a list of choices. You can also use the `Choices...` button to make your selection from a list of available alternatives.

   Defects/Features Identifier for the defect or feature being addressed. For example, you may want to list fix records for specific defects and features.

   Releases Release that the fix records address. For example, you may want to list information on fix records that are associated with the `debugger2` release.

   Components Components to which the fix records refer. For example, you may want to list information on fix records that are referenced by the `code` component.

   States Particular state or states of fix records. Valid states are `notReady`, `ready`, `active`, or `complete`. For example, you may want to list fix records that are in the `ready` state.

   Owner IDs User IDs of the owners of the fix records. For example, you may want to list the fix records that `patjones` owns.

   References Optionally assigned values that group related defects and features.

   Add dates Dates (yy/mm/dd) on which the fix records were created. For example, you may want to list only those fix records created after `93/04/29`. Valid SQL operators for this field are `<`, `>`, `like`, `null`, and `between`.

   Change dates Dates (yy/mm/dd) on which the fix records were last changed. For example, you may want to list only those fix records changed before `93/12/29`. Valid SQL operators for this field are `<`, `>`, `like`, `null`, and `between`.

   Abstracts Summary of the defect or feature that the fix records address. For example, you may want to list all of the fix records dealing with defects or features having the keyword `resource` in them. You could do this by using the wildcard character (`%`) along with the SQL `like` operator and searching for the string `resource`.
3. Select `OK` to perform the query. The CMVC - Fix Records window is refreshed, and the information that you requested is displayed in the client area.

You can use the `Refresh now` item on the File menu to reissue a query to the server. The most recent query is sent and the list is updated with current information from the server. If you issue a query to the server using the `Open list...` option, you can replace the query with the default query specified in the system file by selecting the `Default list` item from the File menu.

Note: For more information about how to use the Options menu items to specify operators, refer to “SQL Operator (in) and No Sort Combo Boxes” on page 11.

Performing Actions with the CMVC - Fix Records Window

This section describes the authority and prerequisites for performing common actions with the CMVC - Fix Records window, and how to perform these actions.

Creating a Fix Record

You can create a fix record to monitor the file changes within a component.

Authority Required: The defect, feature, or track owner has implicit authority. You have explicit authority if the FixCreate action is specified for your user ID in the component associated with the fix record.

Prerequisites: When creating a fix record, you must ensure that the following requirements are met:

- A defect or feature is defined
- Track is in the `fix` state and the fix subprocess is specified for the process of the release associated with the track.

Procedure: To create a fix record:

1. Select `Create...` from the Actions menu to display the Create Fix Records dialog box.

   Defects/Features Identifier of the defect or feature that the track addresses. You can also import a defect or feature from the CMVC - Defects or CMVC - Features windows.

   Releases Release in which the work is being done to address the particular defect or feature. You can also import selected releases from the CMVC - Releases window.

   Component Name of the component that is associated with the files that you want to change. You can also import selected components from the CMVC - Components window.

   Developer User ID of the owner of the fix record. You can also import a selected user ID from the CMVC - Users window.

2. Select `OK` to create the fix record. Each fix record is owned by the component owner unless a developer is specified.

Fix records are created for defects and features according to the accepted sizing records for the defect and feature. They are also created when a file is checked into the CMVC server, if one does not already exist for the component that manages the file.

Completing a Fix Record

You must move the `fix` record to the `complete` state to indicate that all changes required in the associated component are complete.

Authority Required: The fix record, component, or track owner has implicit authority. You have explicit authority if the FixComplete action is specified for your user ID in the component associated with the fix record.

Prerequisites: When completing a fix record, you must ensure that the following requirements are met:

- Track is in the `fix` state
- Fix record is in the `ready` or `active` state

Procedure: To complete a fix record:

1. Select the fix record you want to move to the `complete` state.
2. Select `Complete...` from the Actions menu. A Complete Fix Records dialog box is displayed. Accept the prefilled information in the `Defects/Features`, `Releases`, and `Component` fields or type in new information.
3. Select `OK` to complete the fix record.

When all fix records are in the `complete` state, the track moves to the `integrate`, `commit`, or `complete` states. You can re-activate a completed fix record only if the corresponding track is in the `fix` state.

Note: If you do not want to select the fix records from a list in the client area, you can start with step 2 and type the information directly in the text entry fields.

Activating a Fix Record

You must move the fix record to the `active` state to allow additional file changes to be made after a fix record has been marked complete.

Authority Required: The fix record, component, or track owner has implicit authority. You have explicit authority if the FixActivate action is specified for your user ID in the component associated with the fix record.

Prerequisites: When activating a fix record, you must ensure that the following requirements are met:

- Track is in the `fix` state
- Fix record is in the `ready` or `complete` state.

Procedure: To activate a fix record:

1. Select the fix record you want to move to the `active` state from the list of records in the client area.
2. Select `Activate...` from the Actions menu. An Activate Fix Records dialog box is displayed. Accept the prefilled information in the `Defects/Features`, `Releases`, and `Component` fields or type in new information.
3. Select `OK` to activate the fix record.

Note: If you do not want to select the fix records from a list in the client area, you can start with step 2 and type the information directly in the text entry fields.

Deleting a Fix Record

You can delete a fix record when it is no longer required.

Authority Required: The defect, feature, or track owner has implicit authority. You have explicit authority if the FixDelete action is specified for your user ID in the component associated with the fix record.

Prerequisites: The fix record is not in the `active` or `complete` state.

You cannot delete a fix record that is in the `active` or `complete` state because it has file changes associated with it.

Procedure: To delete a fix record:

1. Select the fix record that you want to delete from the list of records in the client area.
2. Select `Delete...` from the Actions menu. A Delete Fix Records dialog box is displayed. Accept the prefilled information in the `Defects/Features`, `Releases`, and `Component` fields or type in new information.
3. Select `OK` to delete the fix record.

Note: If you do not want to select the fix records from a list in the client area, you can start with step 2 and type the information directly in the text entry fields.

Modifying the Ownership of a Fix Record

You can assign a fix record to another CMVC user ID.

Authority Required: The fix record, component, or track owner has implicit authority. You have explicit authority if the FixAssign action is specified for your user ID in the component associated with the fix record.

Prerequisites: When modifying the ownership of a fix record, you must ensure that the following requirements are met:

- The fix record is not in the `complete` state.
- The new owner has a valid CMVC user ID.

Procedure: To modify the owner of a fix record:

1. Select the fix record for which you want to modify the owner from the list of fix records in the client area.
2. Select `Owner...` from the Modify menu. A Modify Fix Record Owner dialog box is displayed. Accept the prefilled information in the `Defects/Features`, `Releases`, and `Component` fields or type in new information.
3. Type the user ID of the new owner of the fix record in the `New owner` field. You can import a selected user ID from the CMVC - Users window.
4. Select `OK` to modify the owner. The fix record has a new owner who assumes ownership responsibilities. The new owner must complete the fix record once it is in the `active` state.

Note: If you do not want to select the fix records from a list in the client area, you can start with step 2 and type the information directly in the text entry fields.

Showing the Note History of a Fix Record

You can show the notes that are associated with a specific fix record.

Authority Required: None.

Prerequisites: When showing the note history for a fix record, you must ensure that the following requirements are met:

- Fix record exists in CMVC.

- You must have a valid CMVC user ID for the family for which the record is defined.

Procedure: To show the note history for a single fix record, double-click on a fix record listed in the client area.

To show the note history for one or more fix records:

1. Select the fix record for which you want to display the note history from the list of records in the client area.
2. Select `Note history` from the Show menu. The CMVC - Information window is opened and the information that you requested is displayed in the client area.

Showing the Change History of a Fix Record

You can show the history of changes for one or more fix records. For more information about how to show the change history, refer to "Using the CMVC - Change History Window" on page 23.

Test Record Owner's Tasks

Whenever tracks are created for releases that make use of the test process, test records are created according to the environment list of the associated release. Each user on the list is a *tester* for the designated environment, and owns a test record. If the test subprocess is not included in the associated release process, no test records are created.

When the track changes to the `test` state, CMVC informs each tester that the defect or feature change for the related release is ready for testing in its designated environment. The tester must test the release in a specified environment and mark test results on the test record.

As an owner, you can reassign a test record, thus reassigning the role of tester to someone else. Some of the actions that you can perform as a test record owner include:

- Accept or reject a test record
- Abstain from testing
- Modify the ownership of a test record.

Displaying Test Record Information in the Client Area

The CMVC - Test Records window allows you to work with test records. When you query a test record, specific information is displayed in the client area of the window under the following headings:

Prefix Displays the identifier specified when the defect or feature was opened.

Number Displays the number specified when the defect or feature was opened.

Release Displays the release in which the work is being done to address the particular defect or feature.

Environment Displays the environment in which the work done for the particular track needs to be tested.

Owner Displays the user ID of the owner of the test record.

State Displays the current state of the test record. Valid states are `notReady`, `ready`, `accept`, `reject`, or `abstain`.

Reference Displays an optionally assigned value that groups related defects and features.

Created Displays the date and time (yy/mm/dd hh:mm:ss) on which the test record was created.

Updated Displays the date and time (yy/mm/dd hh:mm:ss) on which the test record was last updated.

Abstract Displays a summary of the defect or feature.

Performing Queries with the Open Test Record List Dialog Box

The Open Test Record List dialog box allows you to perform queries and list information in the CMVC - Test Records window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to "Defining Default Queries and Tasks" on page 19.

To perform a query using the Open Test Record List dialog box:

1. Select `Open list...` from the File menu to display the Open Test Record List dialog box.
2. Enter the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

   Prefixes Value that indicates whether the test record refers to a defect or feature. Your family administrator can configure a list of choices. You can use the `Choices...` button to make your selection from a list of available alternatives.

   Defects/Features Value that identifies the defects or features that the test records address. For example, you may want to list only test records for specific defects and features.

   Releases Release for which the test records exist. For example, you may want to list information on the test records that are associated with the `debugger2` release.

   Environments Environment in which the work done for the particular tracks needs to be tested. For example, you may want to list information only on tracks being tested on the `RS_6000`.

   Owner IDs User IDs of the owners of the test records. For example, you may want to list the test records that `patjones` owns.

   States Particular state or states of a test record. Valid values are `notReady`, `ready`, `accept`, `reject` or `abstain`. For example, you may want to list test records that are in the `ready` state.

   References Optionally assigned value that groups related defects or features.

   Add dates Dates (yy/mm/dd) on which the test records were created. For example, you may want to list only the defects created after `93/12/29`. Valid SQL operators for this field are `<`, `>`, `like`, `is null`, and `between`.

   Change dates Dates (yy/mm/dd) on which the test records were last changed. For example, you may want to list test records that were changed between `93/12/14` and `93/09/21/`. Valid SQL operators for this field are `<`, `>`, `like`, `is null`, and `between`.

   Abstracts Summary of the defect or feature that the track addresses. For example, you may want to list all test records for tracks dealing with defects or features having the keyword resources in them. You could do this by using the wildcard character(%) along with the SQL like operator, and search for the string %resource%.

3. Select OK to perform the query. The CMVC - Test Records window is refreshed, and the information that you requested is displayed in the client area.

You can use the Refresh now item on the File menu to reissue a query to the server. The most recent query is sent and the list is updated with current information from the server. If you issue a query to the server using the Open list... option, you can replace the query with the default query specified in the system file by selecting the Default list item from the File menu.

Note: For more information about how to use the Options menu items to specify operators, refer to "SQL Operator (in) and No Sort Combo Boxes" on page 11.

Performing Actions with the CMVC - Test Records Window

This section describes the authority and prerequisites for performing common actions with the CMVC - Test Records window, and how to perform these actions.

Accepting a Test Record

You can accept a test record to indicate successful test results of the file changes in a specific release and environment.

Authority Required: The test record owner has implicit authority. You have explicit authority if the TestAccept action is specified for your user ID in the component associated with the test record.

Prerequisites: The test record is in the ready state.

Procedure: To accept a test record:

1. Select the test record for which you want to indicate successful environment test results from the list of records in the client area.
2. Select Accept... from the Actions menu. An Accept Test Records dialog box is displayed. Accept the prefilled information in the Defects/Features, Releases, Environments and Tester fields or type in new information.
3. Select OK to accept the test record.

When all test records are in either the accept, reject, or abstain state, then the track moves to the complete state.

Note: If you do not want to select the test records from a list in the client area, you can start with step 2 and type the information directly in the text entry fields.

Rejecting a Test Record

You can reject a test record to indicate unsuccessful test results in a release.

Authority Required: The test record owner has implicit authority. You have explicit authority if the TestReject action is specified for your user ID in the component associated with the test record.

Prerequisites: The test record is in the ready state.

Procedure: To reject a test record:

1. Select the test record for which you want to indicate that the environment test results were unsuccessful from the list of records in the client area.
2. Select Reject... from the Actions menu. A Reject Test Records dialog box is displayed. Accept the prefilled information in the Defects/Features, Releases, Environments and Tester fields or type in new information.
3. Select OK to delete the test record.

When all test records are in either the accept, reject, or abstain state, then the track moves automatically to the complete state.

You can open a defect to track the unsuccessful results.

Note: If you do not want to select the test records from a list in the client area, you can start with step 2 and type the information directly in the text entry fields.

Abstaining from Testing

You can abstain from testing the file changes in a release. In this way, you do not judge the success of the changes.

Authority Required: The test record owner has implicit authority. You have explicit authority if the TestAbstain action is specified for your user ID in the component associated with the test record.

Prerequisites: The test record is in the ready state.

Procedure: To abstain from testing the file changes in a release:

1. Select the test record for which you want to abstain from testing.
2. Select Abstain... from the Actions menu. An Abstain from Test dialog box is displayed. Accept the prefilled information in the Defects/Features, Releases, Environments and Tester fields or type in new information.
3. Select OK to abstain from testing.

When all test records are in either the accept, reject, or abstain state, then the track moves to the complete state.

Note: If you do not want to select the test records from a list in the client area, you can start with step 2 and type the information directly in the text entry fields.

Modifying the Ownership of Test Records

You can assign a test record to another CMVC user ID, thus transferring the role of tester.

Authority Required: The test record owner has implicit authority. You have explicit authority if the TestAssign action is specified for your user ID in the component associated with the test record.

Prerequisites: When modifying the ownership of test records, you must ensure that the following requirements are met:

- The new owner has a valid CMVC user ID.
- The test record is in either the `ready` or `notReady` state.

Procedure: To modify the ownership of a test record:

1. Select the test record for which you want to modify the owner from the list of records in the client area.
2. Select `Owner...` from the Modify menu. A Modify Test Record Owner dialog box is displayed. Accept the prefilled information in the `Defects/Features`, `Releases`, `Environments` and `Tester` fields or type in new information.
3. Enter the user ID of the person to whom you want to assign the test records in the `New tester` field. You can also import a selected user ID from the CMVC - Users window.
4. Select `OK` to modify the owner. The test record has a new owner who must mark the test record with either `accept`, `reject`, or `abstain`.

This does not change the environment list associated with the release. For more information about changing environment lists, refer to Chapter 12, “Working with Releases” on page 173.

Note: If you do not want to select the test records from a list in the client area, you can start with step 2 and type the information directly in the text entry fields.

Showing the Note History of a Test Record

You can show the notes that are associated with a specific test record.

Authority Required: None.

Prerequisites: When showing the note history for an approval record, you must ensure that the following requirements are met:

- The test record exists in CMVC.
- You must have a valid CMVC user ID for the family for which the record is defined.

Procedure: To show the note history for a single test record, double-click on a test record listed in the client area.

To show the note history for one or more test records:

1. Select the test record for which you want to display the note history from the list of records in the client area.
2. Select `Note history` from the Show menu. The CMVC - Information window is opened and the information that you requested is displayed in the client area.

Showing the Change History of a Test Record

You can show the history of changes for one or more test records. For more information about how to show the change history, refer to “Using the CMVC - Change History Window” on page 23.

Chapter 10. Working with User IDs and Host Lists

You can use the CMVC - Users window and the CMVC - Host Lists window to perform certain actions on user IDs and host lists. This chapter describes the actions that you can perform as a superuser or owner of a user ID.

Superuser's Tasks

A user with superuser privilege can perform any action. There are some tasks, however, that *only* a superuser can perform. They are:

- Creating or deleting a user ID
- Adding the initial client access on a host for a new user ID or for someone other than themselves
- Removing client access from a host for someone other than themselves.

Displaying User Information in the Client Area

The CMVC - Users window allows you to create and maintain CMVC user identifiers provided that you have superuser authority. When you query a user ID, information is displayed in the client area of the window under the following headings:

User ID Displays the CMVC user ID.

User's Name Displays the full name of the owner of the user ID.

Area Displays the department or area in which the user ID owner works.

User's Mail Address Displays the mailing address in the format *login@hostName*.

Superuser Indicates whether or not the user ID has CMVC superuser privilege.

Created Displays the date and time (yy/mm/dd hh:mm:ss) on which the user ID was created.

Deleted Displays the date and time (yy/mm/dd hh:mm:ss) on which the user ID was deleted.

Updated Displays the date and time (yy/mm/dd hh:mm:ss) on which the most recent changes were made to the information associated with the user ID.

Note: If your family administrator has configured additional fields, other fields may be displayed.

Performing Queries with the Open User List Dialog Box

The Open User List dialog box allows you to perform queries and list information in the CMVC - Users window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to “Defining Default Queries and Tasks” on page 19.

To perform a query using the Open User List dialog box:

1. Select `Open list...` from the File menu to display the Open User List dialog box.
2. Enter the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

   User IDs CMVC user IDs that you want to list. For example, you may want to see the user IDs for `pattut` and `tedtut`.

   User names Full name of the user. For example, you may want to see the user IDs that are owned by Bob Henderson and Jimmy Henries so you could search for the strings `%Hen%` using the `like` SQL operator.

   User areas Department or area in which the user works. For example, you may want to see the user IDs that are in the `Accounting` department.

   User mail addresses Mailing address, in the format *login@hostName*, where the user ID owner wants CMVC mail to be sent.

   Add dates Date (yy/mm/dd) on which the user ID was created. For example, you may want to search for user IDs that were created after `93/01/01`. Valid SQL operators for this field are `<`, `>`, `like`, `is null`, and `between`.

   Delete dates Date (yy/mm/dd) on which the user ID was deleted. For example, you may want to search for user IDs that were deleted after `93/12/01`. Valid SQL operators for this field are `<`, `>`, `like`, `is null`, and `between`.

   Change dates Date (yy/mm/dd) on which the user ID or the information that is associated with the user IDs was last changed. Examples are `chrisb` or `temp103`. You may want to search for user IDs that were created after `93/12/01`. Valid SQL operators for this field are `<`, `>`, `like`, `is null`, and `between`.

3. Type information in any additional fields that have been configured by your family administrator.
4. Select one of the `Superuser` buttons to indicate the scope of your search.

   Yes Search only user IDs with superuser privilege

   No Search only user IDs without superuser privilege

   Both Search user IDs with and without superuser privilege.

5. Select `OK` to perform the query. The CMVC - Users window is refreshed, and the information that you requested is displayed in the client area.

You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to "Defining Default Queries and Tasks" on page 19.

Performing Actions with the CMVC - Users Window

This section describes the authority and prerequisites for performing common actions with the CMVC - Users window, and how to perform these actions.

Creating a User ID

A superuser can create a CMVC user ID for a user.

Authority Required: Only the superuser has implicit authority.

Prerequisites: The user must have a valid *login* on the host system, such as, an OS/2, AIX, HP-UX**, or SunOS** system.

Procedure: To create a user ID:

1. Select `Create...` from the Actions menu to display the Create User dialog box.
2. Type information in the following fields:

   User ID A new user ID. Usually you will make it identical to the user's login. You can type up to 15 alphanumeric characters. You can use all characters except vertical bars (|), ASCII control characters or shell metacharacters.

   User's mail address Mailing address in the format *login@hostName*. You can type up to 143 alphanumeric characters. Examples are `03@hdqtrs.usa` or `psanders@accounts.cntr.one`

   User's full name User's name, surname, and initials as on a business card. You can enter up to 31 characters, including blanks. Examples are `Chris Baker` or `Pat E. Sanders`. Do not use a vertical bar (|), $, ",`, or \, because these have a special meaning to the shell.

   User's area Department, project code, or whatever is appropriate for your organization. Examples are `Accounting` or `Development` or `Maintenance`. You can type up to 15 alphanumeric characters. You can use all characters except vertical bars (|), ASCII control characters or shell metacharacters.

3. Type the required information in any additional fields that have been configured by your family administrator.
4. Select `Grant superuser privilege` if you want to grant CMVC superuser privilege to the user ID.
5. Select `OK` to create the user ID.

The user can receive mail through CMVC at the user's electronic mail address.

A superuser must also make a host list entry for the user ID to have client access. The user ID can issue commands only after the initial host list entry is made.

Deleting a User ID

Only a superuser can delete a CMVC user ID.

Authority Required: Only the superuser has implicit authority.

Prerequisites: The user ID cannot be the owner of any CMVC objects, such as, releases, components, files, sizing records.

Procedure: To delete a user ID:

1. Select the user ID you want to delete from the list of user IDs in the client area.
2. Select `Delete...` from the Actions menu to display the Delete Users dialog box. Accept the prefilled information in the `User IDs` field or type new information.
3. Select `OK` to delete the user ID.

A deleted user ID remains in the system and can be recreated by a superuser only. For more information about recreating an ID, refer to “Recreating a User ID.” It can be listed in the client area and will show the date the user ID was deleted. The deleted user ID, however, cannot access the CMVC server.

Note: If you do not want to select the user ID from a list in the client area, you can begin with step 2 and type the information directly in the text entry field.

Recreating a User ID

You can recreate a CMVC user ID that was deleted.

Authority Required: Only the superuser has implicit authority.

Prerequisites: The user ID must have existed in CMVC and been previously deleted.

Procedure: To recreate a user ID:

1. Select the user ID you want to recreate from the list of user IDs in the client area.
2. Select `Recreate...` from the Actions menu to display the Recreate Users dialog box. Accept the prefilled information in the `User IDs` field or type new information.
3. Select `OK` to recreate the user ID.

A superuser has to create the first host list entry for the recreated user ID to give the user client access to the CMVC server. For more information about adding access, refer to “Adding Client Access to a Host” on page 150.

Note: If you do not want to select the user IDs from a list in the client area, you can begin with step 2 and type the information directly in the User IDs field.

Adding a Host List Entry

A superuser can add a host list entry by selecting the `Add host...` option from the Actions menu. For more information about adding an entry, refer to “Adding a Host List Entry” on page 153.

Displaying Host List Information in the Client Area

The CMVC - Host Lists window allows superusers to change client access for a user ID on a particular host.

Once the superuser adds the host list entry, all CMVC users can:

- Add client access to their host list
- Remove host list entries from their own user IDs.

Note: Only a superuser can remove a host list entry from someone else's user ID.

When you query a host list, information is displayed in the client area of the window under the following headings:

Login Displays the login from which the user can access CMVC.

Host Name Displays the name of the host system from which the user can perform CMVC actions.

User ID Displays the user ID that CMVC recognizes as a valid CMVC user ID. Typically, this is the same as the user's login.

User's Name Displays the full name of the person who owns the user ID.

Area Displays the department or area in which the owner of the user ID works.

Performing Queries with the Open Host List Dialog Box

The Open Host List dialog box allows you to perform queries and list information in the CMVC - Host Lists window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to "Defining Default Queries and Tasks" on page 19.

To perform a query using the Open Host List dialog box:

1. Select `Open list...` from the File menu to display the Open Host List dialog box.
2. Enter the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

   Logins Login identifier required to access CMVC. For example, if you have more than one login, you may want to list the host list entries for all of your logins.

   Host names Host system on which CMVC actions can be performed. For example, you may want to list the host list entries for `prince.new.com` to see the user IDs that can perform CMVC actions from that host system.

   User IDs CMVC user ID for which you want to see the host list entries. For example, you may want to list the host list entries for the user IDs `pat` and `ted`.

User names Full name of the user. For example, you may want to list the host list entries for a user whose surname is `Henries`. You can do this by searching for the string `%Henries%` using the `like` SQL operator.

User areas Department or area in which the owner of the user ID works. For example, you may want to see the host list entries for the user IDs that are owned by people working in `dept45c`.

3. Select `OK` to perform the query. The CMVC - Host Lists window is refreshed, and the information that you requested is displayed in the client area.

You can use the `Refresh now` item on the File menu to reissue a query to the server. The most recent query is sent and the list is updated with current information from the server. If you issue a query to the server using the `Open list...` option, you can replace the query with the default query specified in the system file by selecting the `Default list` item from the File menu.

Note: For more information about how to use the Options menu items to specify operators, refer to “SQL Operator (in) and No Sort Combo Boxes” on page 11.

You can also show the host lists of user IDs by selecting the user IDs from the client area of the CMVC - Users window. See “Showing the Host Lists of User IDs” on page 153 for more information.

Adding Client Access to a Host

You can associate a user ID with a host to give client access.

Authority Required: Only a superuser has implicit authority.

Prerequisites: The user ID exists in CMVC.

Procedure: To add client access to a host:

1. Select `Add...` from the Actions menu to display the Add Host dialog box.
2. Type the information in the following fields:

   Login Login identifier required to access CMVC.

   Host name Name of the host system from which the user ID will access the CMVC server.

   User ID CMVC user ID for which you want to create client access. You can use the Import button to import a selected user ID from the client area of the CMVC - Users window.

3. Select `OK` to add the client access. The user ID can now issue CMVC commands from the host system.

The owner of the user ID can add extra host list entries to have client access from more than one host (see “Adding a Host List Entry” on page 153).

Removing Client Access from a Host

You can remove a host list entry to remove client access from a specific host.

Authority Required: Only a superuser has implicit authority.

Prerequisites: When removing client access from a host, you must ensure that the following requirements are met:

- User ID exists in CMVC
- User ID has one or more host list entries.

Warning: If you remove all host list entries for a CMVC user ID, that user ID will no longer have client access. Only a superuser can add a host list entry to reestablish client access for that user ID.

Procedure: To remove client access from a host:

1. Select the host list entry you want to remove from the list of hosts in the client area.
2. Select `Remove...` from the Actions menu. A Remove Host dialog box is displayed. Accept the prefilled information in the `Login`, `Host name`, and `User ID` fields or type in new information.
3. Select `OK` to remove client access for the user ID.

For more information about re-establishing client access, refer to “Adding Client Access to a Host” on page 150.

Note: If you do not want to select the user ID from a list in the client area, you can begin with step 2 and type the information directly in the text entry field.

User ID Owner’s Tasks

A superuser must create the first user ID for a user and must make the first host list entry to give that user ID host access. After that, the owner of the user ID has implicit authority to:

- Modify a user ID, and the properties of a user ID
- Show details of the user IDs
- Add or remove host list entries.

Modifying a User ID

You can change your user ID.

Authority Required: The user ID owner has implicit authority only.

Prerequisites: The CMVC user ID exists in CMVC.

Procedure: To modify a user ID:

1. Select the user ID you want to modify from the list of user IDs in the client area.
2. Select `User ID...` from the Modify menu in the client area of the CMVC - Users window to display the Modify User IDs dialog box. Accept the prefilled information in the `User ID` field or type new information.

3. Type a new user ID in the `New user ID` field. Examples are `cbaker` or `pes123`. You can type up to 15 alphanumeric characters. You can use all characters except vertical bars (|), ASCII control characters or shell metacharacters.

4. Select `OK` to modify the user ID.

Note: If you do not want to select the user ID from a list in the client area, you can begin with step 2 and type the information directly in the text entry field.

Modifying the Properties of a User ID

You can modify the information associated with your user ID including your:

- Electronic mail address
- Full name
- Area or department
- Superuser privilege (if you have superuser privilege).

Authority Required: The user ID owner has implicit authority.

Note: You must have superuser privilege to change the superuser privilege of a user ID.

Prerequisites: The CMVC user ID exists in CMVC.

Procedure: To modify the properties of a user ID:

1. Select the user IDs you want to modify from the list of user IDs in the client area.

2. Select `Properties...` from the Modify menu in the CMVC - Users window to display the Modify User Properties dialog box. Accept the prefilled information in the `User ID` field or type new information.

3. Type new information in the following fields:

   User's mail address A login and host name in the format *login@hostName*. You can type up to 143 alphanumeric characters. Examples are `vm142_pr1@hdqtrs.xyz` or `psanders@psanders.dev.xyz`.

   User's full name A user's name, surname, and initials. Examples are `Chris Baker` or `Pat E. Sanders`. You can enter up to 31 characters. Do not use a vertical bar (|), `$`, ",`, or \, as these have a special meaning to the shell.

   User's area: A department, project code, or whatever is appropriate for your organization. Examples are `Accounting` or `Development` or `Maintenance`. You can type up to 15 alphanumeric characters. You can use all characters except vertical bars (|), ASCII control characters or shell metacharacters.

4. Type the required information in any additional fields that have been configured by your family administrator.

5. Select one of the `User Privilege` buttons if you want to modify the superuser privilege of the user ID.

   - No change (this is the default).
   - Grant superuser privilege.
   - Revoke superuser privilege.

6. Select `OK` to modify the properties. The properties are changed.

Note: If you do not want to select user IDs from a list in the client area, you can begin with step 2 and then type the information directly in the text entry fields.

Showing the Details of a User ID

You can show information about user IDs, including the following information:

- Name, area, and mailing address of the owner
- Date the user ID was created
- Delete and modify dates, if any.

Authority Required: None.

Prerequisites: The user ID exists in CMVC.

Procedure To show details for a single user ID, double click on a user ID listed in the client area.***:*** To show details for more than one user ID:

1. Select the user IDs from the list of user IDs in the client area of the CMVC - Users window.
2. Select `Details` from the Show menu. The CMVC - Information window is opened and the details of the user IDs are displayed.

Note: If you do not want to select user IDs from a list in the client area, you can begin with step 2 and then type the information directly in the User ID field in the Show User Details dialog box.

Showing the Host Lists of User IDs

You can show the host lists for one or more user IDs, including the login name, host name, user's name, and area.

Authority Required: None.

Prerequisites: The user ID exists in CMVC.

Procedure To show the host lists for one or more user IDs:

1. Select the user IDs from the list of user IDs in the client area of the CMVC - Users window.
2. Select `Host lists` from the Show menu. The CMVC - Host Lists window is opened and the host lists of the user IDs are displayed in the client area.

You can also show the host lists of user IDs by selecting `Open list...` from the File menu of the CMVC - Host Lists window. See “Performing Queries with the Open Host List Dialog Box” on page 149 for the detailed instructions.

Adding a Host List Entry

You can make additional host list entries for your user ID to give yourself client access on more than one host.

Authority Required: The user ID owner has implicit authority.

Prerequisites: When adding host list entries, you must ensure that the following requirements are met:

- A superuser must make the first host list entry for any user ID.
- The host list entry that is being added must not already exist as a host list member for this user ID.

Procedure: To add a host list entry:

1. Select `Add...` from the Actions menu in the CMVC - Host Lists window to display the Add Host dialog box. Accept the prefilled information in the `Login`, `Host name`, and `User ID` fields or type new information.
2. Type information in the following text entry fields:

   Login Login identifier that you use to access CMVC.

   Host name Name of the host system from which you want to access CMVC.

   User ID CMVC user ID for which you want to create client access. You can import a selected user ID from the client area of the CMVC - Users window.
3. Select `OK` to add the entry. You can now issue CMVC commands from the additional host or hosts.

Removing a Host List Entry

You can remove a host list entry to remove client access for a host.

Authority Required: The user ID owner has implicit authority.

Note: You must have superuser privilege to remove a host list entry for another user's ID.

Prerequisites: When deleting a host list entry, you must ensure that the following requirements are met:

- User ID has a host list entry
- Host list entry being removed must exist for the user ID.

Warning: If you remove all host list entries for a CMVC user ID, that user ID will no longer have client access. Only a superuser can add a host list entry to re-establish client access for that user ID.

Procedure: To remove a host list entry:

1. Select the host list entry you want to remove from the list of entries in the client area of the CMVC - Host Lists window.
2. Select `Remove...` from the Actions menu. A Remove Host dialog box is displayed. Accept the prefilled information in the `Login`, `Host name`, and `User ID` fields or type in new information.
3. Select `OK` to remove the host list entry.

Note: If you do not want to select a host list entry from the client area, you can start from step 2 and type the necessary information in the text entry fields.

Chapter 11. Working with Components

The CMVC - Components window enables you to perform a variety of actions on components and to display a list view of information. This chapter describes the actions that you can perform as a component owner with the CMVC - Components window. If you have superuser authority you can perform any of the actions described in this chapter.

Component Owner's Tasks

As a component owner:

- You have sole ownership of a component.
- You can take part in creating and modifying the component structure.
- You are responsible for managing all development data related to a component, including granting access and defining the notification list.

Some of the actions that you can perform as a component owner include:

- Create, delete, or rename a component
- Recreate a deleted component
- Link and unlink a component to and from a parent component
- Open a defect or feature
- Modify the owner or description of a component
- Show the details or process configuration of a component
- Add or remove entries in the access list and notification list
- Restrict access to a component
- Show the actions that a CMVC user can perform.

When a defect or feature is opened against a component, the component owner becomes a defect owner or a feature owner, by default. For more information about the tasks of a defect owner and feature owner, refer to Chapter 4, "Working with Defects" on page 45 and Chapter 5, "Working with Features" on page 65 respectively.

A component owner has implicit authority over files associated with a component. For more information about the tasks that are related to files, refer to Chapter 3, "Working with Files" on page 25.

Listing Component Information in the CMVC - Components Window

The CMVC - Components window allows you to work with components. When you query a component, specific information is displayed in the client area of the window under the following headings:

Component Displays the name of the component.

Owner Displays the user ID of the component owner.

Owner's Name Displays the full name of the component owner.

Area Displays the department or area in which the component owner works.

Process Displays the name of the process that applies to this component. Your family administrator can specify the CMVC subprocesses that are included in the component process.

Created Displays the date and time (yy/mm/dd hh:mm:ss) on which the component was created.

Deleted Displays the date and time (yy/mm/dd hh:mm:ss) on which the component was deleted.

Updated Displays the date and time (yy/mm/dd hh:mm:ss) on which the component was updated.

Description Displays the information that describes the purpose of the component.

Performing Queries with the Open Component List Dialog Box

The Open Component List dialog box allows you to perform queries and list information in the CMVC - Components window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to "Defining Default Queries and Tasks" on page 19.

To perform a query using the Open Component List dialog box:

1. Select `Open list...` from the File menu to display the Open Component List dialog box.
2. Enter the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

   Components Name of the component. For example, you may want to list information on components containing the word tools in their name. You could do this by searching for the string `%tools%` using the SQL `like` operator.

   Owner IDs User IDs of the owner of the component. For example, you may want to list the components having `pattut` as the owner.

   Owner names Full name of the component owner. For example you may want to list the components that are owned by `Bob Henderson`. You can search for the string `%Henderson%` using the `like` SQL operator.

   Owner areas Department or area in which the component owner works. For example, you may want to list the components that are owned by users in `groupABC`.

   Process Type the name of the process for this component. This field can accept up to 15 characters. You can use the `Choices...` button to make your selection from a list of available alternatives.

   Add dates Dates (yy/mm/dd) on which the components were added. For example, you may want to search for components added since `93/12/01`. Valid SQL operators for this field are <, >, `like`, `is null`, and `between`.

Delete dates Dates (yy/mm/dd) on which the components were deleted. For example, you may want to search for components deleted since `93/12/01`. Valid SQL operators for this field are <, >, `like`, `is null`, and `between`.

Change dates Dates (yy/mm/dd) on which the components were last changed. For example, you may want to search for components updated since `93/12/01`. Valid SQL operators for this field are <, >, `like`, `is null`, and `between`.

Descriptions A description of the purpose of the component. For example, you may want to list all components dealing with the administration of projects. You do this by searching for the string `%admin%` using the `like` SQL operator.

3. Select `OK` to perform the query. The CMVC - Components window is refreshed, and the information that you requested is displayed in the client area.

You can use the `Refresh now` item on the File menu to reissue a query to the server. The most recent query is sent and the list is updated with current information from the server. If you issue a query to the server using the `Open list...` option, you can replace the query with the default query specified in the system file by selecting the `Default list` item from the File menu.

Note: For more information about how to use the Options menu items to specify operators, refer to "SQL Operator (in) and No Sort Combo Boxes" on page 11.

Performing Actions with the CMVC - Components Window

This section describes the authority and prerequisites for performing common actions with the CMVC - Components window, and how to perform these actions.

Creating a Component

You can create a component to organize data into manageable groups and to control the user access and notification.

Authority Required: The owner of the *parent component* has implicit authority. You have explicit authority if the CompCreate action is specified for your user ID in the access list of the parent component.

Prerequisites: The parent component exists.

Procedure: To create a component:

1. Select `Create...` on the Actions menu to display the Create Components dialog box.
2. Type information in the following fields:

Components Unique names for each of the components you are creating. Examples are `project1` or `admin1` or `account1`. You cannot use the name of an existing component in your family or one that was previously deleted. You can type up to 31 alphanumeric characters. You can use all characters except vertical bars (|), ASCII control characters or shell metacharacters.

Parent Name of an existing component. This component will be the parent of the new components.

Process Type the name of the process for this component. You can type up to 15 alphanumeric characters. You can use the `Choices...` button to make your selection from a list of available alternatives.

Owner User ID of the person who is to be the owner of the component you are creating. If you do not specify an owner, then you will be the owner of the new component by default. You can use the `Import` button to pull in a selected user ID from the client area of the CMVC - Users window.

Description Purpose of the component. You can type up to 63 alphanumeric characters. For example, `Project 1 administration` or `Documents for Project 1`. Do not use a vertical bar (|), `$0`, ”,`, or \, because these have a special meaning to the shell.

3. Select `OK` to create the component. A new component is added to the hierarchy. This component inherits all access list entries and notification list entries from the parent component.

If you specified a component owner other than yourself, you should give that owner explicit access authority to this component so that access can be given to other users. For more information about granting access authority and notification, refer to “Adding Access List Entries” on page 166 and refer to “Adding Notification List Entries” on page 171.

Linking a Component to a Parent

You can rearrange the structure of the component hierarchy in your family by linking components to some components and unlinking them from others. The linking action precedes the unlinking action because a component must have at least one parent.

Authority Required: The component owner of a component being linked has implicit authority. You have explicit authority if the CompLink action is specified for your user ID in the component that is being linked.

Prerequisites: When linking a component to a parent, you must ensure that the following requirements are met:

- Components exist in CMVC.
- Components are not marked as deleted.
- Link between the components is unidirectional. For example, only one component in a pair can be the parent component.

Note: A component can be the parent to many child components, and a component can have many parents.

Procedure: To link one or more components to a parent:

1. Select the component that you want to link to another parent component from the list of components in the client area.

2. Select Link... on the Actions menu to display the Link Components dialog box. Accept the prefilled information in the Components field or type new information.
3. Type the name of the existing component that is to be the additional parent in the Parent field. This cannot be a child component or a descendant of the component you are linking.
4. Select OK to link the components. The component now has more than one parent component. It inherits the superset of all authority groups and notification groups granted at parent and ancestor components.

Note: If you do not want to select the component from a list in the client area, you can begin with step 2 and type the information directly into the entry fields.

Unlinking a Child Component from a Parent

You can rearrange the structure of the component hierarchy in your family by linking and unlinking components. The linking action precedes the unlinking action because a component must have at least one parent.

Authority Required: The component owner of the component being unlinked has implicit authority. You have explicit authority if the CompUnlink action is specified for your user ID in the component that is being unlinked.

Prerequisites: When unlinking a component from another component, you must ensure that the following requirements are met:

- Components are created in CMVC.
- Components are not marked as deleted.
- Link between the components is unidirectional. For example, only one component in a pair can be the parent component.
- The component that you are unlinking from has at least one other parent component.

Procedure: To unlink a component from another component:

1. Select the component you want to unlink from a parent component from the list of components in the client area.
2. Select Unlink... from the Actions menu to display the Unlink Components dialog box. Accept the prefilled information in the Components field or type new information.
3. Type the name of the parent component from which you want to unlink the component in the Parent field.
4. Select OK to unlink the component. The component is no longer related to the parent component and does not inherit the access and notification lists of the parent from which it was unlinked.

Note: If you do not want to select the component from a list in the client area, you can begin with step 2 and type the information directly into the entry fields.

Deleting a Component

You can delete a component so that it is no longer active in the component hierarchy. This means that you are prevented from opening defects or features for the component and assigning objects to the component. If you want to restore the component, you can recreate it.

Authority Required: The component owner has implicit authority. You have explicit authority if the CompDelete action is specified for your user ID in the component being deleted.

Prerequisites: When deleting a component, you must ensure that the following requirements are met:

- The components are not marked as deleted.
- The component that you want to delete is not related to any files, child components, releases, open features, open defects, or active sizing records.

  Note: You must reassign or delete any such objects to other components before performing the deletion.
- You *cannot* delete the root component.

Procedure: To delete a component:

1. Select the component you want to delete from the list of components in the client area.
2. Select `Delete...` from the Actions menu to display the Delete Components dialog box. Accept the prefilled information in the `Components` field or type new information.
3. Select `OK` to delete the component. The component is deleted. The access lists and notification lists associated with that component are also deleted.

If you delete a component by mistake, you can recreate it. For more information about recreating a component, refer to “Recreating a Deleted Component.”

Deleted components still appear in the client area. You can reuse them by recreating and renaming them for your current needs.

Note: If you do not want to select the components from a list in the client area, you can begin with step 2 and then enter the information in the `Components` field.

Recreating a Deleted Component

If you delete a component by mistake, you can recreate the component.

Authority Required: The component owner of a new parent component has implicit authority. You have explicit authority if the CompRecreate action is specified for your user ID in the new parent component.

Prerequisites: The component exists in CMVC but has been deleted.

Procedure: To recreate a component:

1. Select the component that you want to recreate from the list of components in the client area.

2. Select Recreate... from the Actions menu to display the Recreate Components dialog box. Accept the prefilled information in the Components field or type new information.

3. Type the name of the parent component in the Parent field.

4. Select OK to recreate the component. You become the owner of the recreated component by default.

For more information about reassigning the component to another user ID, refer to "Modifying Ownership of a Component" on page 163. The access and notification lists are not recreated.

If you retain ownership of the component then you can create new lists. If you reassign ownership of the component, then the new owner of the recreated component can create new lists. For more information about access and notification lists, refer to "Adding Access List Entries" on page 166 and "Adding Notification List Entries" on page 171, respectively.

Opening a Defect

You can open a defect by selecting the Open Defect item on the Actions menu. For more information about opening a defect, refer to "Opening a Defect" on page 49.

Opening a Feature

You open a feature by selecting the Open Feature item from the Actions menu. For more information about opening features, refer to "Opening a Feature" on page 68.

Adding an Access List Entry

You can add a user ID to an access list.

Authority Required: The release owner has implicit authority. You have explicit authority if the AccessCreate action is specified for your user ID in the component associated with the release.

Prerequisites: When adding an access list entry, you must ensure that the following requirements are met:

- The release exists in CMVC.
- The approval subprocess is included in the release process.

Procedure: To add access list entries:

1. Select Add access... from the Actions menu to display the Add Access dialog box.

2. Type information in the following fields:

   Component Name of an existing component for which you want to make an access list entry.

   Users User IDs for which you want to make the access list entry. You can also import a selected user ID from the CMVC - User window.

   Authority Predefined access authority group. You can use the Choices... push button to make your selection from a list of available alternatives.

3. Select `OK` to add the user ID. The user IDs are added to the access list for that release.

Adding a Notification List Entry

You can add a user ID to a notification list.

Authority Required: The release owner has implicit authority. You have explicit authority if the NotifyCreate action is specified for your user ID in the component associated with the release.

Prerequisites: The release exists in CMVC.

Procedure: To add a notification list entry:

1. Select `Add notification...` from the Actions menu to display the Add Notification dialog box.
2. Type information in the following fields:

   Component Component for which you want to make a notification list entry.

   Users Used IDs for which you want to make the notification list entry. You can also import a selected user ID from the CMVC - Users window.

   Interest A predefined notification interest group. You can use the `Choices...` push button to make your selection from a list of alternatives.
3. Select `OK` to add the entry. The user ID is added to the notification list for that release.

Renaming a Component

You can rename a component.

Authority Required: The component owner has implicit authority. You have explicit authority if the CompModify action is specified for your user ID in the component being renamed.

Prerequisites: When renaming a component, you must ensure that the following requirements are met:

- The component exists in CMVC.
- The component is not marked as deleted.

Procedure: To rename a component:

1. Select the component you want to rename from the list of components in the client area.
2. Select `Name...` from the Modify menu to display the Modify Component Name dialog box. Accept the prefilled information in the `Components` field or type new information.
3. Type a new name for the component in the `New name` field. You can type up to 31 alphanumeric characters. You can use all characters except vertical bars (|), ASCII control characters or shell metacharacters. You cannot use the name of an existing component in your family nor one that was previously deleted.
4. Select `OK` to rename the component.

Note: If you do not want to select the component from a list in the client area, you can begin with step 2 and type the information directly into the entry fields.

Modifying Ownership of a Component

You can change the owner of a component.

Authority Required: The component owner has implicit authority. You have explicit authority if the CompModify action is specified for your user ID in the component being reassigned to another component.

Prerequisites: The component exists in CMVC.

You can assign the ownership of a component to another user ID.

Procedure: To assign the ownership of a component to another user ID:

1. Select the component that you want to assign to a new owner.
2. Select `Owner...` from the Modify menu to display the Modify Component Owner dialog box. Accept the prefilled information in the `Components` field or type new information.
3. Type the user ID of the new owner in the `New owner` field. You can use the Import button to import a selected user ID from the client area of the CMVC - Users window.
4. Select `OK` to modify the owner.

Note: If you do not want to select the component from a list in the client area, you can begin with step 2 and then type the information directly in the text entry fields.

Modifying the Properties of a Component

You can modify component properties, such as, the description.

Authority Required: The component owner has implicit authority. You have explicit authority if the CompModify action is specified for your user ID in the component that is being modified.

Prerequisites: When modifying the component properties, you must ensure that the following requirements are met:

- Component exists in CMVC
- Component is not marked as deleted.

Procedure: To modify the component properties:

1. Select the component for which you want to change the properties from the list of components in the client area.
2. Select `Properties...` from the Modify menu to display the Modify Component Properties dialog box. Accept the prefilled information in the `Components` field or type new information.
3. Type new information in the following fields:

   Process Type the name of the process for the component. You can type up to 15 alphanumeric characters. You can use the `Choices...` button to make your selection from a list of available alternatives.

Description Type the description for the component. You can type up to 63 alphanumeric characters. Do not use a vertical bar (|), $, ",`, or \, because these have a special meaning to the shell.

4. Select `OK` to modify the component properties.

Note: If you do not want to select the component from a list in the client area, you can begin with step 2 and then type the information directly in the text entry fields.

Showing the Component Details

You can show the details of a component, including the:

- Component name
- Component owner
- Date on which the component was created
- Date on which the component was last modified, (if applicable) or deleted
- Parent components
- Child components, including all descendants
- Process being used by the component

Authority Required: The component owner has implicit authority. You have explicit authority if the CompView action is specified for your user ID in the component that is being displayed.

Prerequisites: The component exists in CMVC.

Procedure: To show the details of a single component, double-click on a component listed in the client area.

To show details for more than one component:

1. Select the components for which you want to show the details from the list of components in the client area.
2. Select `Details` from the Show menu. The CMVC - Information window is opened and the details about the components are displayed.

Note: If you do not want to select the component from a list in the client area, you can begin with step 2 and then type the information directly in the `Component` field in the Component Details dialog box.

Showing the Process Configurations for a Component

You can show the process being used by the component.

Authority Required: The component owner has implicit authority. You have explicit authority if the CompView action is specified for your user ID in the component that is being displayed.

Prerequisites: The component exists in CMVC.

Procedure: To show information for one or more components:

1. Select the components for which you want to show the process from the list of components in the client area.
2. Select `Process configurations` from the Show menu. The CMVC - Information window is opened and the process configurations for the components are displayed.

Note: If you do not want to select the component from a list in the client area, you can begin with step 2 and then type the information directly in the `Components` field in the Show Component Process Configuration dialog box.

Additional Show Options

Additional show options allow you to open certain GUI windows, including the CMVC - Access Lists, CMVC - Notification Lists, CMVC - Defects, CMVC - Features, CMVC - Files, and CMVC - Releases windows. To open these windows, select the corresponding option from the Show menu.

Working with Access Lists

An access list identifies the users who can access the associated component and the actions or commands that those users are authorized to complete. Access list entries are inherited from all parent and ancestor components.

Use the CMVC - Access Lists window to create and maintain access lists for the components. You can also use the CMVC - Components window to create access lists for the components.

There is a default set of authority groups that are shipped with CMVC, but this set can be configured by the CMVC family administrator. For more information about the authority groups and the actions each group is authorized to perform, refer to the *IBM CMVC User's Reference*.

Displaying Access List Information in the Client Area

The CMVC - Access Lists window allows you to work with access lists. When you query an access list entry, information is displayed in the client area of the window. The information is displayed under the following headings:

Component Displays the name of the component for which the access list was created.

User ID Displays the user ID of the person who has access defined for a component.

Authority Displays the explicit authority group that action is specified for a component.

Type Displays the type of access authority. Valid types are granted and restricted.

User's Name Displays the full name of the user who has access defined for a component.

Area Displays the department or area in which the user works.

Performing Queries with the Open Access Lists Dialog Box

The Open Access List dialog box allows you to perform queries and list information in the CMVC - Access Lists window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to "Defining Default Queries and Tasks" on page 19.

To perform a query using the Open Access List dialog box:

1. Select `Open list...` from the File menu to display the Open Access List dialog box.
2. Enter the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

   Components Names of the components for which you want to list access list entries. For example, you may want to list access list entries for components containing the term `tool` in their name. You could do this by using the wildcard (`%`) along with the SQL `like` operator and searching for the string `%tools%`.

   User IDs User IDs that are given explicit access through the access list entries for the component. For example, you may want to list the access list entries for the tester `pattut`.

   Authorities Access authority groups that are defined for the user IDs. For example, you may want to list who has developer access for various components. You could do this by searching for the string `developer` using the `in` SQL operator. You can also use the `Choices...` button to make your selection from a list of available alternatives.

   Types The access authority type. Valid types are granted or restricted.

   User names Full names of the users with access list entries. For example, you may want to list the access that Bob Henderson has for various components. You could do this by searching for the string `%Henderson%` using the `like` SQL operator.

   User areas Department or area in which the users work. For example, you may want to list the access list entries for the users in `dept4`.
3. Select `OK` to perform the query. The CMVC - Access Lists window is refreshed, and the information that you requested is displayed in the client area.

You can use the `Refresh now` item on the File menu to reissue a query to the server. The most recent query is sent and the list is updated with current information from the server. If you issue a query to the server using the `Open list...` option, you can replace the query with the default query specified in the system file by selecting the `Default list` item from the File menu.

Note: For more information about how to use the Options menu items to specify operators, refer to “SQL Operator (in) and No Sort Combo Boxes” on page 11.

Performing Actions with the CMVC - Access Lists Window

This section describes the authority and prerequisites for performing common actions with the CMVC - Access Lists window, and how to perform these actions.

Adding Access List Entries

You can add access for a user by associating their user ID with a preconfigured authority group.

Existing authority groups can be modified and new ones can be defined by your family administrator.

Authority Required: The component owner has implicit authority. You have explicit authority if the AccessCreate action is specified for your user ID in the component where access is being added.

Prerequisites: You cannot grant another user any access authority that is not defined for your own user ID.

Procedure: To add an access list entry:

1. Select `Add...` from the Actions menu of the CMVC - Access Lists window to display the Add Access dialog box.
2. Type the following information in the text entry fields:

   Component Name of an existing component for which you want to make an access list entry. You can also use the `Import` button to pull in a selected component name from the client area of the CMVC - Components window.

   Users User IDs for which you want to make the access list entries. You can also use the Import button to pull in selected user IDs from the client area of the CMVC - Users window.

   Authority A predefined access authority group. You can also use the `Choices...` button to make your selection from a list of available alternatives.
3. Select `OK` to add the entry. The access is added at the component, and the user inherits this access at all child components.

Restricting Access at a Component

You can restrict access for a specific component and indicate whether this restriction is for all inherited users or only specific inherited users.

Authority Required: The component owner has implicit authority. You have explicit authority if the AccessRestrict action is specified for your user ID in the component where the access is being restricted.

Note: Restricted authority does not affect a user with implicit or superuser authority.

Prerequisites: When restricting access authority for a user ID, you must ensure that the following requirements are met:

- Component exists in CMVC
- User does not have granted or restricted access type established at the component for the particular authority group that is being restricted.

Procedure: To restrict access:

1. Select `Restrict...` from the Actions menu of the CMVC - Access Lists window to display the Restrict Access dialog box.
2. Type the appropriate information in the following fields:

   Component Name of an existing component for which you want to restrict access. You can also import a selected component from the CMVC - Components window.

Users User IDs for which you want to restrict access for a specific component. You can also import a selected user ID from the CMVC - Users window.

Authority Access authority you want to restrict for the group. Use the `Choices...` push button to select from a list of available alternatives.

3. To restrict access for all users, select `All users`.
4. Select `OK` to restrict the authority. Access authority is restricted for the specific users for the component. The users whose access has been restricted are notified. If all inherited users are restricted at the component, only those users who have AccessRestrict authority will be notified.

Note: Restricted access authority is not inherited.

To remove the restricted authority, follow the procedures in the section “Removing Access List Entries.”

Removing Access List Entries

You can remove access for a user by removing one or more access list entries.

Authority Required: The component owner has implicit authority. You have explicit authority if the AccessDelete action is specified for your user ID in the component where the access is being deleted.

Prerequisites: The access list entry exists.

Procedure: To remove access for a user:

1. Select the access list entries you want to remove from the list of entries in the client area of the CMVC - Access Lists window.
2. Select `Remove...` from the Actions menu. A Remove Access dialog box is displayed. The information for the `Component` and `Users` fields is already filled in.
3. Select `OK` to remove access to the specified component.
4. To remove access for all users, select `All users`.

When you remove an access list entry, the user no longer inherits that access at all child components. The user can get the owners of one or more child components to add access list entries if such access is necessary.

Note: If you do not want to select the component from a list in the client area, you can begin with step 2 and then type the information directly in the text entry fields.

Showing the Actions that a User Can Perform

You can show the actions that a CMVC user can perform on a component.

Authority Required: None.

Prerequisites: None.

Procedure: To show authority actions for an access list entry, double-click on an access list entry listed in the client area.

You can also show the actions a CMVC user can perform by:

1. Select the access list entry for which you want to show the authority actions from the list of entries in the client area.
2. Select `Authority actions` from the Show menu. The CMVC - Information window is opened and the information that you requested is displayed.

Note: If you do not want to select the component from a list in the client area, you can begin with step 2 and then type the information directly in the Authority field in the Show Authority Actions dialog box.

Working with Notification Lists

A notification list is used to inform users whenever an action within their interest group occurs in relation to a given component. All mail is routed to the mailing address that is specified for a user ID. More than one interest group can be associated with a user ID for a given component. Notification list entries are inherited from all parent and ancestor components.

Use the CMVC - Notification Lists window to create and maintain notification lists for the components.

There is a default set of notification interest groups that are shipped with CMVC, but this set can be configured by the CMVC family administrator. For more information about the notification interest groups and the actions of which each group is notified, refer to the *IBM CMVC User's Reference* manual. If you are the family administrator, and you want to redefine the notification interest groups or define some additional groups, you should read the book *IBM CMVC Server Administration and Installation*.

Displaying Notification List Information in the Client Area

The CMVC - Notification Lists window allows you to work with notification lists. When you query a notification list entry, information is displayed in the client area under the following headings:

Component Displays the name of the component for which the notification list was created.

User ID Displays the user ID of the person who has notification interest defined for a component.

Interest Displays the interest group defined for a component for a user ID.

User's name Displays the full name of the user who has notification interest defined for a component.

Area Displays the department or area in which the user works.

User's Mail Address Displays the *login@hostname* address to which CMVC sends a message whenever an action that is specified in the notification interest group of a CMVC user ID occurs.

Performing Queries with the Open Notification List Dialog Box

The Open Notification List dialog box allows you to perform queries and list information in the CMVC - Notification Lists window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to "Defining Default Queries and Tasks" on page 19.

To perform a query using the Open Notification List dialog box:

1. Select `Open list...` from the File menu to display the Open Notification List dialog box.
2. Enter the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

   Components Name of the components for which you want to list notification list entries. For example, you may want to list the notification list entries for the `debugger2` component to list the notification interest that users have at that component.

   User IDs User IDs with notification interest identified at a specific component. For example, you may want to list the notification list entries for the user ID `pattut`.

   Interests Notification interest groups that are defined for your user IDs. For example, you may want to list who has high notification interest at various components. You can also use the `Choices...` button to make your selection from a list of available alternatives.

   User names Full name of the users with notification interest defined for a component. For example, you may want to list the notification list entries that Bob Henderson has for various components. You could do this by searching for the string `%Henderson%` using the `like` SQL operator.

   User areas Department or area in which the users work. For example, you want to list the notification list entries for the users in `dept4`.

   User mail addresses Mail address to which CMVC sends a message, in the format *login@hostname*. For example, you may want to search for the notification list entries that involve the mailing address `peter@prod2.xyz.com`.

3. Select `OK` to perform the query. The CMVC - Notification Lists window is refreshed, and the information that you requested is displayed in the client area.

You can use the `Refresh now` item on the File menu to reissue a query to the server. The most recent query is sent and the list is updated with current information from the server. If you issue a query to the server using the `Open list...` option, you can replace the query with the default query specified in the system file by selecting the `Default list` item from the File menu.

Note: For more information about how to use the Options menu items to specify operators, refer to “SQL Operator (in) and No Sort Combo Boxes” on page 11.

Performing Actions with the CMVC - Notification Lists Window

This section describes the authority and prerequisites for performing common actions with the CMVC - Notification Lists window, and how to perform these actions.

Adding Notification List Entries

Users who own a CMVC object are automatically notified when certain actions are performed or when their action is required. Component owners can explicitly identify subscribers who want to get notified of actions out of interest; actions about which they would not be automatically notified.

You can add users to a notification list if the users are interested in being notified when certain actions are performed. You do this by associating their user ID with a preconfigured interest group.

Existing interest groups can be modified and new ones can be defined by your family administrator.

Authority Required: The component owner has implicit authority. You have explicit authority if the NotifyCreate action is specified for your user ID in the component associated with the notification list.

Prerequisites: None.

Procedure: To add notification list entries:

1. Select `Add...` from the Actions menu of the CMVC - Notification Lists window to display the Add Notification dialog box.
2. Type the appropriate information in the following fields:

   Component Name of an existing component for which you want to make a notification list entry. You can also import a selected component name from the client area of the CMVC - Components window.

   Users User IDs for which you want to make the notification list entry. You can also import selected user IDs from the client area of the CMVC - Users window.

   Interest A predefined notification interest group. You can use the `Choices...` button to make your selection from a list of available alternatives.
3. Select `OK` to add the entry. The notification is added at the component, and the user inherits this notification at all child components.

Removing Notification List Entries

You can remove explicit notification for a user by removing one or more notification list entries.

Note: Automatic notification cannot be turned off.

Authority Required: The component or user ID owner has implicit authority. You have explicit authority if the NotifyDelete action is specified for your user ID in the component associated with the notification list.

Prerequisites: The notification list entry exists.

Procedure: To remove notification list entries:

1. Select the notification list entry that you want to remove from the list of entries in the client area.
2. Select `Remove...` from the Actions menu. A Remove Notification dialog box is displayed. The information for the Component, Users, and Interest fields is already filled in.
3. Select `OK` to remove notification at the component.

In addition, the user no longer inherits that notification at all child components. The user can get the owners of one or more child components to add notification list entries if this is necessary.

Note: If you do not want to select the notification entry from a list in the client area, you can begin with step 2 and then type the information directly in the Component, Users, and Interest fields in the Remove Notification dialog box.

Showing the Interest Actions for a Notification List Entry

You can display information on the actions that are associated with an interest group.

Authority Required: The component owner has implicit authority. You have explicit authority if the NotifyView action is specified for your user ID in the component associated with the notification list.

Prerequisites: The interest group exists.

Procedure: To show the actions associated with an interest group, double-click on an entry listed in the client area.

You can also show the actions for an interest group by:

1. Select the notification list entry for which you want to show the interest actions from the list of entries in the client area.
2. Select `Interest actions` from the Show menu. The CMVC - Information window is opened and the interest actions are displayed.

Note: If you do not want to select an entry from the list in the client area, you can begin with step 2 and type the information directly in the Interest field in the dialog box.

Chapter 12. Working with Releases

The CMVC - Releases window allows you to perform a variety of actions on releases. This chapter describes the actions that the release owner and certain other users can perform. If you have superuser authority, you can perform any task in this chapter.

Release Owner's Tasks

The release owner can use releases to:

- Group files together along a single line of development
- Change the properties of a release.

When the tracking subprocess has been specified for a release process, the release owner controls two main steps in the change control process. They are:

- Deciding whether the release is to include the approval subprocess and, if so, creating a release approver list to indicate the users who must approve proposed changes before the changes are made to the release.
- Deciding whether the release is to include the test subprocess and, if so, creating a release environment list to identify the environments in which the release must be tested and to designate a tester for each environment.

Some of the actions that you can perform as a release owner include:

- Create, delete, or recreate a release
- Extract files for a release
- Link files to a release
- Modify the release name, owner, properties, or component
- Show the details of a release
- Show the process for the release
- Add and remove approver list entries
- Add and remove environment list entries.

Displaying Release Information in the Client Area

The CMVC - Releases window allows you to create and maintain releases in CMVC. When you query a release, information is displayed in the client area of the window under the following headings:

Release Displays the name of the release

Component Displays the name of the component that manages the release

Owner Displays the user ID of the owner of the release

Owner's Name Displays the full name of the owner of the release

Area Displays the department or area in which the owner works

Process Displays the name of the process that applies to this release

Created Displays the date and time (yy/mm/dd hh:mm:ss) the release was created

Deleted Displays the date and time (yy/mm/dd hh:mm:ss) the release was deleted

Updated Displays the date and time (yy/mm/dd hh:mm:ss) that the most recent changes were made to the release

Description Displays the information that describes the purpose of the release.

Performing Queries with the Open Release List Dialog Box

The Open Release List dialog box allows you to perform queries and list information in the CMVC - Releases window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to "Defining Default Queries and Tasks" on page 19.

To perform a query using the Open Release List dialog box:

1. Select `Open list...` from the File menu to display the Open Release List dialog box.
2. Enter the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

   Releases Names of the releases that you want to list. For example, you may want to list all releases containing the term `tool` in their name. You could do this by using the wildcard character (`%`) along with the SQL `like` operator, and searching for the string `%tool%`.

   Components Names of the components that manage the releases. For example, you may want to list information on releases referenced by the `tools` component.

   Owner IDs User IDs of the owners of the releases. For example, you may want to list the releases that have `pattut` as the owner.

   Owner names Full name of the release owner. For example, you may want to list the releases that are owned by the users in `dept4`.

   Owner areas Departments or areas in which the owner works. For example, you may want to list the releases that are owned by the users in `dept4`.

   Processes Name of the process that applies to the release. You can use the `Choices...` button to make your selection from a list of available alternatives.

   Add dates Dates (yy/mm/dd) on which the releases were created. For example, you may want to search for releases added since `93/12/01`. Valid SQL operators for the field are `<`, `>`, `like`, `is null`, and `between`.

   Delete dates Dates (yy/mm/dd) on which the releases were deleted. For example, you may want to search for releases deleted since `93/12/01`. Valid SQL operators for the field are `<`, `>`, `like`, `is null`, and `between`.

Change dates Dates (yy/mm/dd) on which the releases were last changed. For example, you may want to search for releases updated since `93/01/01`. Valid SQL operators for the field are `<`, `>`, `like`, `is null`, and `between`.

Descriptions Description of the purpose of the release. For example, you may want to list all releases for the project `alpha`. You could do this by searching for the string `%alpha%` using the `like` SQL operator.

3. Select `OK` to perform the query. The CMVC - Releases window is refreshed, and the information that you requested is displayed in the client area.

You can use the `Refresh now` item on the File menu to reissue a query to the server. The most recent query is sent and the list is updated with current information from the server. If you issue a query to the server using the `Open list...` option, you can replace the query with the default query specified in the system file by selecting the `Default list` item from the File menu.

Note: For more information about how to use the Options menu items to specify operators, refer to "SQL Operator (in) and No Sort Combo Boxes" on page 11.

Performing Actions with the CMVC - Releases Window

This section describes the authority and prerequisites for performing common actions with the CMVC - Releases window, and how to perform these actions.

Creating a Release

You can create releases to maintain multiple versions of a product.

Authority Required: No one has implicit authority. You have explicit authority if the ReleaseCreate action is specified for your user ID in the component associated with the new releases.

Prerequisites: The component that you want to manage the release must already exist in CMVC.

Procedure: To create a release:

1. Select `Create...` from the Actions menu to display the Create Release dialog box.
2. Type information in the following fields:

Releases Unique name for the release you are creating. Examples are `alpha` or `10debugr` or `graphix21`. You cannot use the name of an existing release in your family nor one that was previously deleted. You can type up to 15 characters. You can use all characters except vertical bars (`|`), ASCII control characters or shell metacharacters.

Component Name of an existing component with which you want to associate the release you are creating. You can also import a selected component from the CMVC - Components window.

Process Name of the process that applies to the release. You can use the `Choices...` button to make your selection from a list of available alternatives.

Owner User ID of the person you want to be the owner of the release. You can also import a selected user ID from the CMVC - Users window.

Description Description of the purpose of the release. You can type up to 63 alphanumeric characters. Examples are `debugger version 1` or `graphics editor version 2.1`. Do not use a vertical bar (|), $, ", `, or \, as these have a special meaning to the shell.

- If the name of the process you specify includes the CMVC test subprocess, you may also be required to type information in one or more of the following fields to create an *environment list*:

  Environment Name of the environment you want to add for purposes of testing a specific release.

  Tester User ID of the person you want to be the tester for the release in these environments.

  This information is added to the environment list for the release.

  Note: Depending on the subprocess that is included in the process for the release, the server will check that information is supplied for all required fields.

- If the name of the process you specify includes the CMVC approval subprocess, you must also enter the user ID of the person who owns the approval record in the `Approver` field. This information is added to the approver list for the release.

3. Select `OK` to create the release. You become the release owner by default unless you explicitly enter another user ID in the `Owner` field.

Deleting a Release

You can delete a release if you no longer need it.

Authority Required: The release owner has implicit authority. You have explicit authority if the ReleaseDelete action is specified for your user ID in the component associated with the release.

Prerequisites: You cannot delete releases if they have:

- Levels that are in the `working` state or the `integrate` state. To check if a release has any levels in the `working` or `integrate` states, you can open a list from the CMVC - Levels window and search for levels by state and release.
- Files associated with them. To check if a release has any files associated with it, you can open a list from the CMVC - Files window and search for files.
- Active tracks. To check if a release has any tracks that are not in the `complete` state, you can open a list from the CMVC - Tracks window and search for tracks.

Procedure: To delete a release:

1. Select the release you want to delete from the list of releases in the client area.
2. Select `Delete...` from the Actions menu to display the Delete Releases dialog box. Accept the prefilled information in the `Releases` field or type new information.
3. Select `OK` to delete the release.

When you delete releases, any environment list and approver list associated with the releases are also deleted.

Note: If you do not want to select the releases from a list in the client area, you can begin with step 2 and then type the information directly in the Releases fields.

Recreating a Release

You can recreate a release after it has been deleted.

Authority Required: The release owner has implicit authority. You have explicit authority if the ReleaseRecreate action is specified for your user ID in the component associated with the release.

Prerequisites: The components with which the releases were associated must still exist.

Procedure: To recreate a release:

1. Select the release that you want to recreate from the list of releases in the client area.
2. Select `Recreate...` from the Actions menu to display the Recreate Release dialog box. Accept the prefilled information in the `Releases` field or type new information.
3. If the release to be recreated is associated with a process that includes the test subprocess, you may also be required to type information in one or more of the following fields to create an environment list:

   Environment Name of the environment you want to add for purposes of testing a specific release.

   Tester User ID of the person you want to be the tester for the release in these environments.

   This information is added to the environment list for the release.

   Note: Depending on the subprocess that is included in the process for the release, the server will check that information is supplied for all required fields.
4. If the release to be recreated is associated with a process that includes the approval subprocess, you must also enter the user ID of the person who owns the approval record in the `Approver` field. This information is added to the approver list for the release.
5. Select `OK` to recreate the release. You become the release owner by default unless you explicitly enter another user ID in the `Owner` field.

When you recreate releases, old environment and approver lists are not recreated. The owner of the recreated releases must create new environment and approver lists.

Note: If you do not want to select the releases from a list in the client area, you can begin with step 2 and then type the information directly in the Releases fields.

Extracting Files for a Release

If you extract the committed version of a release, any base files that are deleted after the release was committed are still included in the extraction. Deleted base files, however, are not included when you request release extracts using versions or base files that are changed after a certain date.

Authority Required: The release owner has implicit authority. You have explicit authority if the ReleaseExtract action is specified for your user ID in the component associated with the release.

Prerequisites: You require an NFS exported directory from the NFS server host.

Procedure: To extract a file for a release:

1. Select the release for which you want to extract files from the list of releases in the client area.
2. Select `Extract...` from the Actions menu to display the Extract Release dialog box. Accept the prefilled information in the `Releases` field or type new information.
3. Indicate the files that you want to extract by clicking on one of the following radio buttons:

   Current version Current version of the releases' files. This is the default setting. Deleted files are not included.

   Committed version Committed version of the releases' files, including any files deleted after the releases were committed.

   Changed after date Files changed after a certain date. Deleted base files are not included.
4. If you want to extract only those files modified after a certain date, type a date (yy/mm/dd) in the `Date` field. Examples are `93/12/29` or `94/01/03`.
5. Type the destination host and directory in the following fields:

   Host A host machine on which to place the extracted files. Examples are `gmh` or `astro`.

   Directory A directory from the host in which to place the extracted file tree. Examples are `/tmp/test/graphix` or `/debugr/src`. This directory must already be exported via NFS.
6. Type optional information regarding file, directory access, and ownership for the extracted file tree in the following fields:

   User number (UID) Internal number that uniquely identifies the user to the system. The default is the family UID. Examples are `200` or `104`.

   Group number (GID) Internal number that uniquely identifies the group to the system. The default is the family GID. Examples are `1` or `4`.

   File permissions Read, write, and execute file permissions in octal number notation. The default is the file mode associated with the file stored on the CMVC server, with read-only access for all users.

Directory permissions Read, write, and execute directory permissions in octal number notation. The default is 750. For example, `750` represents read, write, and execute access for the owner, read and execute access for the others in the owner's group, and no access for all other users.

7. Select `Expand keywords` to indicate that you do not want to substitute assigned values in place of keywords imbedded in the file. The default is to substitute values for the keywords. For a list of supported keywords, refer to the book *IBM CMVC User's Reference*.

8. If your target host is running the OS/2 operating system and you have indicated the target directory using the proper OS/2 file-naming convention, CMVC adds the Carriage Return and Line Feed characters to all extracted files as a default.

   If your target host is running AIX, SunOS, or HP-UX, click on the `CRLF` button to add Carriage Return and Line Feed characters to the extracted files.

9. Select `OK` to extract the files for the release. All files associated with the release (except destroyed files) now exist in the specified host's directory. If duplicate files exist from the host's directory, they are overwritten.

Note: If you do not want to select the releases from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Linking Files in One Release to Another Release

You can link all active files in a release to another release. By doing this you can decide whether you want to link the current or committed version of a file. If you select the current version, all active files are linked. If you select the committed version, the committed version of all active files are linked.

Authority Required: The release owner has implicit authority. You have explicit authority if the ReleaseLink action is specified for your user ID in the component associated with the release from which the files are being linked.

Prerequisites: When linking files to a release, you must ensure that the following requirements are met:

- Release from which you want to link exists in CMVC.
- Release to which you want to link exists in CMVC.
- If the tracking subprocess is included in the process for the new release, a track must exist.

Procedure: To link files belonging to a specified release to a new release:

1. Select one or more releases that you want to link from the list of releases in the client area.
2. Select `Link...` from the Actions menu to display the Link Release dialog box. Accept the prefilled information in the `Releases` field or type new information.
3. Type information in the following fields:

   New Release Name of the new release to which you want to link the existing files in a release.

   Defects/Features Identifiers of the defects and features for the release.

4. Select one of the following buttons to indicate the files that you want to extract:

   Current version Current version of the releases' files. This is the default setting. Deleted files are not included.

   Committed version The committed version of the releases' files including any files deleted after the releases were committed.

   Changed after date The files changed after a certain date. Deleted base files are not included.

5. If you want to extract only those files modified after a certain date, type a date (yy/mm/dd) in the `Date` field. Examples are `93/11/23` or `94/01/02`.
6. Select `OK` to link the file to the new release.

Note: If you do not want to select the releases from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Adding an Approver List Entry

The users listed on the approver list must approve the proposed changes to the release before the changes can be made.

Authority Required: The release owner has implicit authority. You have explicit authority if the ApproverCreate action is specified for your user ID in the component associated with the release.

Prerequisites: When adding an approver list entry, you must ensure that the following requirements are met:

- Release exists in CMVC
- Approval subprocess is specified for the release process.

Procedure: To add approver list entries:

1. Select `Add approvers...` from the Actions menu to display the Add Approvers dialog box.
2. Type information in the following fields:

   Release Name of the release for which you want to specify approvers.

   Users User IDs of the people you want to be approvers for the release. You can also import a selected user ID from the CMVC - Users window.

3. Select `OK` to add the user IDs. The user IDs are added to the approver list for that release.

Note: Modifying a release's approver list does not affect existing approval records.

Adding an Environment List Entry

Persons that are on an environment list must test the implemented feature or resolved defect in the specified environment.

Authority Required: The release owner has implicit authority. You have explicit authority if the EnvCreate action is specified for your user ID in the component associated with the release.

Prerequisites: When adding an environment list entry, you must ensure that the following requirements are met:

- Release exists in CMVC
- Test subprocess is specified for the release process

Procedure: To add an environment list entry:

1. Select `Add environments...` from the Actions menu to display the Add Environments dialog box.
2. Type information in the following fields:

   Environments Names of environments you want to add for purposes of testing a specific release.

   Release Name of an existing release you want to have tested.

   Tester User ID of the person you want to be the tester for the release in these environments. You can also import a selected user ID from the CMVC - Users window.
3. Select `OK` to add the entry. The user ID is added to the environment list for that release.

Modifying the Name of a Release

You can modify the names of releases.

Authority Required: The release owner has implicit authority. You have explicit authority if the ReleaseModify action is specified for your user ID in the component associated with the release.

Prerequisites: When modifying the name of a release, you must ensure that the following requirements are met:

- Release exists in CMVC
- Release is not marked as deleted.

Procedure: To modify the name of a release:

1. Select the release that you want to rename from the list of releases in the client area.
2. Select `Name...` from the Modify menu to display the Modify Release Name dialog box. Accept the prefilled information in the `Releases` field or type new information.
3. Type a new name for the release in the `New name` field. Examples are `alpha` or `10debugr` or `graphix21`. You can type up to 15 characters. You can use all characters except vertical bars (`|`), ASCII control characters or shell metacharacters. You cannot use the name of an existing release in your family nor one that was previously deleted.
4. Select `OK` to modify the release name. The change is reflected everywhere that the releases are referenced.

Note: If you do not want to select the releases from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Modifying the Owner of a Release

You can reassign ownership of a release to another user.

Authority Required: The release owner has implicit authority. You have explicit authority if the ReleaseModify action is specified for your user ID in the component associated with the release.

Prerequisites: The release exists in CMVC.

Procedure: To reassign ownership of a release to another user:

1. Select the release you want to assign to another owner from the list of releases in the client area.
2. Select `Owner...` from the Modify menu to display the Modify Release Owner dialog box. Accept the prefilled information in the `Releases` field or type new information.
3. Type the user ID of the person you want to be the new owner of the release in the `New owner` field. You can also import a selected user ID from the CMVC - Users window.
4. Select `OK` to assign the release to another owner. The new owner assumes the ownership responsibilities of the release.

Note: If you do not want to select the releases from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Modifying the Component that Manages a Release

You can modify the component that manages a release.

Authority Required: The release owner has implicit authority. You have explicit authority if the ReleaseCreate and ReleaseModify actions are specified for your user ID in the component associated with the release.

Prerequisites: When modifying the component that manages a release, you must ensure that the following requirements are met:

- The release exists in CMVC and is not marked as deleted
- The new component exists and is not marked as deleted.

Procedure: To modify the component that manages a release:

1. Select the release you want to assign to another component from the list of releases in the client area.
2. Select `Component...` from the Modify menu to display the Modify Release Component dialog box. Accept the prefilled information in the `Releases` field or type new information.
3. Type the name of an existing component you want to associate with the release (replacing the previous component) in the `New component` field. You can also import a selected component from the CMVC - Components window.
4. Select `OK` to change the component.

Note: If you do not want to select the releases from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Modifying the Properties of a Release

You can modify certain properties of a release, such as the description, or a process.

Authority Required: The release owner has implicit authority. You have explicit authority if the ReleaseModify action is specified for your user ID in the component associated with the release.

Prerequisites: The release exists in CMVC.

Procedure: To modify a property of a release:

1. Select the release for which you want to modify the properties from the list of releases in the client area.
2. Select Properties.. from the Modify menu to display the Modify Release Properties dialog box. Accept the prefilled information in the Releases field or type new information.
3. You can type a new description for the release using the following fields:

   Process Name of the process that applies to the release. You can use the Choices... button to make your selection from a list of available alternatives.

   Description Information about this release. Examples are debugging editor version 1. You can type up to 63 alphanumeric characters. Do not use a vertical bar (|), $, ", `, or \, because these have a special meaning to the shell.

4. If the process you specify includes the test subprocess, you must also enter information in the following fields to create a test environment.

   Environment Name of the environment you want to add for purposes of testing a specific release.

   Tester User ID of the person you want to be the tester for the release in these environments

   This information is added to the environment list for the release.

5. If the process you specify includes the approval subprocess, you must also type the user ID of the person who owns the approval record in the Approver field. This information is added to the approver list for the release.
6. Select OK to modify the release. Tracks created prior to this modification are not changed.

Note: If you do not want to select the releases from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Showing the Details of a Release

You can view information about a release, including the:

- Name of the release
- Owner
- Description
- Process name and subprocesses associated with the release
- Various create and modify dates.

Authority Required: The release owner has implicit authority. You have explicit authority if the ReleaseView action is specified for your user ID in the component associated with the release.

Prerequisites: The release exists in CMVC.

Procedure: To show information for a single release, double-click on a release listed in the client area.

To show information for more than one release:

1. Select the releases for which you want to show the release details from the list of releases in the client area.
2. Select `Details` from the Show menu. The CMVC - Information window is opened and the details are displayed in the client area.

Note: If you do not want to select the releases from a list in the client area, you can begin with step 2 and then type the information directly in the `Releases` field.

Showing the Process Configurations for a Release

You can show the process that is configured for the release.

Authority Required: The release owner has implicit authority. You have explicit authority if the ReleaseView action is specified for your user ID in the component associated with the release.

Prerequisites: The release exists in CMVC.

Procedure: To show process information for one or more releases:

1. Select the releases for which you want to show the release process from the list of releases in the client area.
2. Select `Process configurations` from the Show menu. The CMVC - Information window is opened and the process configurations are displayed in the client area.

Note: If you do not want to select the releases from a list in the client area, you can begin with step 2 and then type the information directly in the Releases field in the Show Release Process Configurations dialog box.

Showing the Locked Files for a Release

You can show the files that are locked for the release.

Authority Required: None.

Prerequisites: The release exists in CMVC.

Procedure: To show the files that are locked for one or more releases:

1. Select the releases for which you want to show the locked files from the list of releases in the client area.
2. Select `Files locked` from the Show menu. The CMVC - Information window is opened and the locked files are displayed in the client area.

Note: If you do not want to select the releases from a list in the client area, you can begin with step 2 and then type the information directly in the `Releases` field.

Additional Show Options

Additional show options allow you to open certain GUI windows, including the CMVC - Environment Lists, CMVC - Approver Lists, CMVC - Files, CMVC - Tracks, and CMVC - Levels windows. To open these windows, select the corresponding option from the Show menu.

Working with Release Approver Lists

Use the CMVC - Approver Lists window to add entries to or remove entries from the release approver list.

The owners of the user IDs on the approver list of a release must approve the proposed changes to the release before the changes can be made. The approver's decision must be marked on their approval record. For more information, refer to Chapter 9, “Working with Approval, Fix, and Test Records” on page 125.

Displaying Approver List Information in the Client Area

The CMVC - Approver List window allows you to work with approver lists. When you query an approver list entry, information is displayed in the client area of the window under the following headings:

Release Displays the name of the release for which the approver list was created.

Approver Displays the user ID of the person who approves the changes for the release.

Approver's Name Displays the full name of the person who approves changes for the release.

Area Displays the department or area in which the approver works.

Performing Queries with the Open Approver List Dialog Box

The Open Approver List dialog box allows you to perform queries and list information in the CMVC - Approver Lists window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to “Defining Default Queries and Tasks” on page 19.

To perform a query using the Open Approver List dialog box:

1. Select `Open list...` from the File menu to display the Open Approver List dialog box.
2. Enter the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

Releases Name of the releases for which you want to list approver list entries. For example, you may want to list approver list entries for all releases containing tool in their name. You could do this by using the wildcard (`%`) along with the SQL `like` operator and searching for the string `%tool%`.

User IDs User IDs of the approvers. For example, you may want to list the approver list entries for tester `pattut`.

User names Full name of the approvers. For example, you may want to list the approver list entries for `Bob Henderson`. You could do this by searching for the string `%Henderson%` using the `like` SQL operator.

User areas Department or area in which the approvers work. For example, you may want to list the approver list entries for the users in `dept4`.

3. Select `OK` to perform the query. The CMVC - Approver Lists window is refreshed, and the information that you requested is displayed in the client area.

You can use the `Refresh now` item on the File menu to reissue a query to the server. The most recent query is sent and the list is updated with current information from the server. If you issue a query to the server using the `Open list...` option, you can replace the query with the default query specified in the system file by selecting the `Default list` item from the File menu.

Note: For more information about how to use the Options menu items to specify operators, refer to “SQL Operator (in) and No Sort Combo Boxes” on page 11.

Performing Actions with the CMVC - Approver Lists Window

This section describes the authority and prerequisites for performing common actions with the CMVC - Approver Lists window, and how to perform these actions.

Adding an Approver List Entry

The users listed on the approver list must approve the proposed changes to the release before the changes can be made.

Authority Required: The release owner has implicit authority. You have explicit authority if the ApproverCreate action is specified for your user ID in the component associated with the release.

Prerequisites: The release exists in CMVC.

Procedure: To add approver list entries:

1. Select `Add...` from the Actions menu to display the Add Approvers dialog box.
2. Type information in the following fields:

Release Name of the release for which you want to specify approvers. You can also import a selected release from the CMVC - Releases window.

Users User IDs of the people you want to be approvers for the release. You can also import a selected user ID from the CMVC - Users window.

3. Select `OK` to add the user IDs. The user IDs are added to the approver list for that release.

Note: Modifying a release's approver list does not affect existing approval records.

Removing an Approver List Entry

You remove a person from an approver list if you no longer require them to approve the proposed changes to a release.

Authority Required: The release owner has implicit authority. You have explicit authority if the ApproverDelete action is specified for your user ID in the component associated with the release.

Prerequisites: When removing an approver list entry, you must ensure that the following requirements are met:

- Release exists in CMVC
- Entry on the approver list that you want to delete exists in CMVC.

Procedure: To remove an approver list entry:

1. Select the approver list entry you want to remove from the approver list in the client area.
2. Select `Remove...` from the Actions menu. A Remove Approvers dialog box is displayed. Accept the prefilled information in the `Release` and `Users` fields or type in new information.
3. Select `OK` to remove the approve list entry.

Modifying a release's approver list does not affect existing approval records. When you recreate a release, old approver list entries are not recreated. The owner of the recreated release must create new approver list entries.

Note: If you do not want to select the releases from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Working with Release Environment Lists

Use the CMVC - Environment Lists window to add, remove, and modify entries for a particular release.

An environment list specifies the environments in which an implemented feature or resolved defect must be tested for a given release. Each entry in the list must consist of an environment in which testing has to be done, and the user ID of the designated tester. Each tester must enter the results of the testing on a test record.

Displaying Environment List Information in the Client Area

The CMVC - Environment Lists window allows you to create and maintain environment lists in CMVC. When you query a release, information is displayed in the client area of the window under the following headings:

Environment Displays the environment in which testing for a release is required.

Release Displays the name of the release for which the environment was created.

Tester Displays the user ID of the person who performs the testing in the specified environment and release.

Tester's Name Displays the full name of the tester.

Area Displays the department or area in which the tester works.

Performing Queries with the Open Environment List Dialog Box

The Open Environment List dialog box allows you to perform queries and list information in the CMVC - Environment Lists window. You can also use it to define a query that can be made into a default query for the window or added to the Tasks list in the CMVC - Tasks window.

For more information about how to use the Open List dialog box and the `Default query` option to define a default query or add a task to the Tasks list, refer to "Defining Default Queries and Tasks" on page 19.

To perform a query using the Open Environment List dialog box:

1. Select `Open list...` from the File menu to display the Open Environment List dialog box.
2. Enter the search criteria, along with any SQL operators to refine the query, in one or more of the following fields:

   Environments Environments in which testing for a release is required. For example, you may want to list environment list entries for the `micro` environment.

   Releases Name of the releases for which you want to list environment list entries. For example, you may want to list the environment list entries for the `debugger2` release.

   User IDs User IDs of the testers. For example, you may want to list the approver list entries for tester `pattut`.

   User names Full name of the approvers. For example, you may want to list the approver list entries for `Bob Henderson`. You could do this by searching for the string `%Henderson%` using the `like` SQL operator.

   User areas Department or area in which the approvers work. For example, you may want to list the approver list entries for the users in `dept4`.
3. Select `OK` to perform the query. The CMVC - Environment Lists window is refreshed, and the information that you requested is displayed in the client area.

You can use the `Refresh now` item on the File menu to reissue a query to the server. The most recent query is sent and the list is updated with current information from the server. If you issue a query to the server using the `Open list...` option, you can replace the query with the default query specified in the system file by selecting the `Default list` item from the File menu.

Note: For more information about how to use the Options menu items to specify operators, refer to "SQL Operator (in) and No Sort Combo Boxes" on page 11.

Performing Actions with the CMVC - Environment Lists Window

This section describes the authority and prerequisites for performing common actions with the CMVC - Environment Lists window, and how to perform these actions.

Adding an Environment List Entry

Persons that are on an environment list must test the implemented feature or resolved defect in the specified environment.

Authority Required: The release owner has implicit authority. You have explicit authority if the EnvCreate action is specified for your user ID in the component associated with the release.

Prerequisites: The release exists in CMVC.

Procedure: To add an environment list entry*colon.

1. Select `Add...` from the Actions menu to display the Add Environments dialog box.
2. Type information in the following fields:

   Environments Names of environments you want to add for purposes of testing a specific release. You can type up to 15 alphanumeric characters.

   Release Name of an existing release you want to have tested. You can also import a selected release from the CMVC - Releases window.

   Tester User ID of the person you want to be the tester for the release in these environments. You can also import a selected user ID from the CMVC - Users window.

3. Select `OK` to add the entry. The user ID is added to the environment list for that release.

Removing an Environment List Entry

You remove a person from an environment list if you no longer require them to test the release in the specified environment.

Authority Required: The release owner has implicit authority. You have explicit authority if the EnvDelete action is specified for your user ID in the component associated with the release.

Prerequisites: When removing an environment list entry, you must ensure that the following requirements are met:

- Release exists in CMVC
- Environment list entry that you wish to delete exists in CMVC.

Procedure: To remove an environment list entry:

1. Select the environment list entries you want to remove from the list of entries in the client area.
2. Select `Remove...` from the Actions menu. A Remove Environments dialog box is displayed. Accept the prefilled information in the `Environments` and `Release` fields or type in new information.
3. Select `OK` to remove an environment list entry.

Modifying a release's environment list does not affect existing environment list entries. When you recreate a release, old environment list entries are not recreated. The owner of the recreated release must create new environment list entries.

Note: If you do not want to select the releases from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Modifying an Environment List Tester

You can change the name of the tester from the environment list.

Authority Required: The release owner has implicit authority. You have explicit authority if the EnvModify action is specified for your user ID in the component associated with the release.

Prerequisites: When modifying an environment list tester, you must ensure that the following requirements are met:

- Release exists in CMVC.
- There is at least one entry in the environment list.
- Entry that you want to modify exists in the environment list.

Procedure: To change the name of the tester on the environment list:

1. Select the environment list entry for which you want to change the tester from the list of entries in the client area.
2. Select `Tester...` from the Modify menu to display the Modify Environment Tester dialog box. Accept the prefilled information in the `Environments` and `Release` fields or type in new information.
3. Type the user ID of the new tester of the specified environments in this release in the New tester field. You can also import a selected user ID from the CMVC - Users window.
4. Select `OK` to change the entries. The environment list entries for that release are modified.

Note: If you do not want to select the releases from a list in the client area, you can begin with step 2 and then type the information directly in the entry fields.

Glossary

Glossary terms are defined as they are used in this book. If you cannot find the term for which you are looking, refer to the index or the *IBM Dictionary of Computing*, SC20-1699.

A

absolute path name. A directory or a file expressed as a sequence of directories followed by a file name beginning from the root directory.

access list. A CMVC object that controls access to development data. A list of user ID-authority group pairs attached to a component, designating users and the corresponding authority access they are being granted for all objects managed by this component or any of its descendants. It also contains the user ID-authority group pairs designating users who are restricted from performing actions at a specific component.

action. A task performed by the CMVC server and requested by a CMVC client. A CMVC action is the same as issuing one CMVC command.

approval record. A status record on which an approver must give an opinion of the proposed file changes required to resolve a defect or implement a feature in a release.

approver. A user who approves changes within a specific release.

approver list. A list of user IDs attached to a release, representing the users who must approve file changes required to resolve a defect or implement a feature in that release.

authority. The right to access development objects and perform CMVC commands. See also *access list, explicit authority, implicit authority, restricted authority* and *superuser privilege*.

B

base authority. The set of actions granted to a user when a user ID is created within a CMVC family. Contrast with *implicit authority* and *explicit authority*.

base file name. The name assigned to the file outside of the CMVC server environment, excluding any directory names.

C

change control. The process of limiting and auditing changes to files through the mechanism of checking files in and out of a central, controlled, storage location. Change control for individual releases can be integrated with problem tracking by specifying a process for the release that includes the tracking subprocess.

check in. The return of a CMVC file to version control.

check out. The retrieval of a revision of a CMVC file from version control.

child component. All components in each CMVC family, except the root component, are created in reference to an existing component. The existing component is the parent component, and the new component is the child component. A parent component can have more than one child component. See also *component, parent component*.

client. A workstation that requests services from another workstation.

CMVC client. A workstation with the CMVC client software installed.

CMVC file. A file that is stored by the CMVC server and retrieved by a path name. See also *file, common file,* and *shared file*.

CMVC server. A workstation with the CMVC server software installed.

command. A request to perform an operation or run a program from the command line interface. In CMVC, a command consists of the command name, one action flag, and zero or more attribute flags.

common file. A file that is in two or more releases. The same version of the file is the current version for those releases. See also *shared file*.

comparison operator. An operator used in comparison expressions, such as, > (greater than), < (less than), > = (greater than or equal to), < = (less than or equal to), and = (equal to).

component. A CMVC object that organizes project data into structured groups, and controls configuration management properties. Component owners can control access to development data and configure notification about CMVC actions. Components are in a parent-child hierarchy, with descendent components inheriting access and notification information from ancestor components. See also *notification list* and *access list*.

configuration management. The process of identifying, managing, and controlling software modules as they change over time.

context. A description of a data file or directory in the form *host dir file*. That is the host machine, working directory, and file. See *tool context* and *message context*.

corequisite tracks. Two or more tracks designated as corequisites by a user so that all tracks in the corequisite group must be included as members in the same level. If a track is added to a level, all tracks that have a corequisite relationship with that track must also be included in the level before the level is committed.

current working directory. The directory that is the starting point for relative path names; the directory in which you are working.

D

database. A systematized collection of data that can be accessed and operated upon by a data processing system for a specific purpose.

default. A value that is used when an alternative is not specified by the user.

default query. A database search, defined for a specific CMVC GUI window, that is issued each time the CMVC GUI window is opened. See also *search*.

defect. A CMVC object used to formally report a problem. The user who opens a defect is the defect originator.

delete. If you delete a development object, such as, a file or a user id, any reference to that object is removed from CMVC. Certain objects can be deleted only if certain criteria are met. Most objects that are deleted can be re-created.

delta file tree. A directory structure representing only the files that are changed for tracks in a specified level.

destroy. The only CMVC development object that can be destroyed is a file. Destroying a file removes the file record from the database on the CMVC server. Though a destroyed file cannot be re-created, it is included as part of an extracted level.

directory file list. A list of files and sub-directories of the current working directory displayed in the Development Manager main window.

E

environment. A user-defined testing domain for a release. Also used as a defect field, in which case it is the environment where the problem occurred.

environment list. A CMVC object used to specify environments in which a release should be tested. A list of environment-user ID pairs attached to a release, representing the user responsible for testing each environment. Only one tester can be identified for an environment.

explicit authority. The ability to perform an action against a CMVC object because you have been granted the authority to perform that action. Contrast with *implicit authority* and *base authority*.

extract. A CMVC action you can perform on a file, level, or release. A file extraction results in the specified file being copied to the client workstation. A level extraction and release extraction result in the files for the level or release being copied to a designated NFS server.

F

family. A logical organization of related development data. A single CMVC server can support multiple families. The data in one family cannot be accessed from another family.

family administrator. A user who is responsible for all nonsystem related tasks for one or more CMVC families, such as, planning, configuring, and maintaining the CMVC environment and managing user access to those families.

feature. A CMVC object used to formally request and record information about a functional addition or enhancement. The user who opens a feature is the feature originator.

file. A collection of data that is stored by the CMVC server and retrieved by a path name. Any text or binary file used in a development project can be created as a CMVC file. Examples include source code, executables, documentation, and test cases. See *common file* and *shared file*.

fix record. A status record that is associated with a track and is used to monitor the phases of change within each component that is affected by a defect or feature for a specific release.

G

GID. Group number.

graphical user interface (GUI). The OSF/Motif-based CMVC graphical user interface program.

group number (GID). A number that uniquely identifies the files' group to the system.

GUI. Graphical user interface.

H

host. A host node, host computer, or host system.

host list. A list associated with each CMVC user ID that indicates the client hosts that can access the CMVC server and act on behalf of the CMVC user. The CMVC server uses the list to authenticate the identity of a CMVC client upon receipt of a CMVC command. Each entry consists of a login, a CMVC user ID, and a host name.

I

implicit authority. The ability to perform an action on a CMVC object without being granted explicit authority. This authority is implicitly granted through inheritance or object ownership. Contrast with *explicit authority*.

import. Bring selected items to a field in a dialog box from a matching main CMVC GUI window.

integrated problem tracking. The process of integrating problem tracking with change control to track all reported defects, all proposed features, and all subsequent changes to files. See also *change control*.

L

level. A collection of tracks that represent a set of changed files within a release. Levels are only associated with releases whose processes include the track and level subprocesses.

level member. A track that is added to a level.

lock. An action that prevents editing access to a file stored in the CMVC development environment so that only one user can change a file at a time.

login. Operating system user identification.

M

map. The process of reassigning the meaning of an object.

N

Network File System (NFS). A program that allows you to share files with other computers in one or more networks over a variety of machine types and operating systems.

NFS. Network File System.

notification list. A CMVC object allowing component owners to configure notification. A list of user ID-interest group pairs attached to a component, designating users and the corresponding notification interest they are being granted for all objects managed by this component or any of its descendants.

O

operator. A symbol that represents an operation to be done. See *comparison operators*.

originator. The user who opens a defect or feature and is responsible for verifying the outcome of the defect or feature on a verification record. This responsibility can be reassigned.

owner. The user who is responsible for a CMVC object within a CMVC family, either because the user created the object or was assigned ownership of the object.

P

parent component. All components in each CMVC family, except the root component, are created in reference to an existing component. The existing component is the parent component. See also *child component* and *component*.

path name. The name of the file under CMVC control. A path name can be a set of directory names and a base name or just a base name. It must be unique within the release that groups the files.

prerequisite tracks. If a file is changed to resolve more than one defect or feature, the track referenced by the first change is a prerequisite of the track referenced by later changes. A track is a prerequisite to another track if:

- File changes are checked in, but not committed, for the first track.

- One or more of the same files are then checked out, changed, and checked in again for the second track.

process. A combination of CMVC subprocesses, configured by the family administrator, that controls the general movement of CMVC objects (defects, features, tracks, and levels) from state to state within a component or release. See also *subprocess* and *state*.

Q

query. A structured request for information from a database, for example, a search for all defects that are in the open state. See also *default query* and *search*.

R

relative path name. The name of a directory or a file expressed as a sequence of directories followed by a file name, beginning from the current directory.

release. A CMVC object defined by a user that groups all the files that must be built, tested, and distributed as a single entity.

restricted authority. The restriction of a user's ability to perform certain actions at a specific component. Authority can be restricted by the superuser, the component owner, or a user with AccessRestrict authority. See also *authority*.

root component. The initial component that is created when a CMVC family is configured. All components in a CMVC family are descendants of the root component. Only the root component has no parent component. See also *component, child component,* and *parent component*.

S

scope. A parameter in the TOOL statement for each SDE WorkBench/6000 or HP SoftBench tool. It defines the fields in the message context used by the Execution Manager to determine whether a tool can service a request.

search. The scanning of one or more data elements of a set in a database to find elements that have certain properties.

server. A workstation that performs a service for another workstation.

shared file. A file that is shared between two or more releases. See also *common file*.

shell script. A series of commands combined in a file that carry out a function when the file is run.

sizing record. A status record created for each component-release pair affected by a proposed defect or feature. The sizing record owner must indicate whether the defect or feature affects the specified component-release pair and the approximate amount of work needed to resolve the defect or implement the feature within the specified component-release pair.

state. Tracks, levels, features, and defects move through various states during their life cycles. The state of an object determines the actions that can be performed on it. See also *process* and *subprocess*.

subprocess. CMVC subprocesses govern the state changes for CMVC objects. The design, size, review (DSR) and verify subprocesses are configured for component processes. The track, approve, fix, level, and test subprocesses are configured for release processes. See also *process* and *state*.

superuser privilege. This privilege allows a user to perform any action available in the CMVC family.

Note: Superuser privilege is internal to CMVC and not related to operating system superuser authority.

T

tester. A user responsible for testing the resolution of a defect or the implementation of a feature for a specific level of a release and recording the results on a test record.

test record. A status record used to record the outcome of an environment test performed for a defect or feature in a level of a release.

track. A CMVC object created to monitor the progress of changes made to files within a release to resolve a specific defect or to implement a specific feature.

track subprocess. An attribute of a CMVC release process that specifies that the change control process for that release will be integrated with the problem tracking process.

U

UID. User number.

user. A person with an active CMVC user ID and access to one or more CMVC families.

user number (UID). A number that uniquely identifies a login on a specific host and controls the ownership of a file within the file system.

V

verification record. A status record that must be marked by the originator of a defect or a feature before the defect or feature can move to the closed state. Originators can use verification records to verify the resolution or implementation of the defect or feature they opened.

version control. The storage of different versions of a file and information about each version.

W

working file. The checked-out version of a CMVC file.

Index

Printed in U.S.A.

SC09-1783-00