

# Vault End User Guide

---

Optegra<sup>®</sup> Release 6

DOC40167-007

---

**Copyright © 2001 Parametric Technology Corporation. All Rights Reserved.**

User documentation from Parametric Technology Corporation (PTC) is subject to copyright laws of the United States and other countries and is provided under a license agreement, which restricts copying, disclosure, and use of such documentation. PTC hereby grants to the licensed user the right to make copies in printed form of PTC user documentation provided on software or documentation media, but only for internal, noncommercial use by the licensed user in accordance with the license agreement under which the applicable software and documentation are licensed. Any copy made hereunder shall include the Parametric Technology Corporation copyright notice and any other proprietary notice provided by PTC. User documentation may not be disclosed, transferred, or modified without the prior written consent of PTC and no authorization is granted to make copies for such purposes.

Information described in this document is furnished for general information only, is subject to change without notice, and should not be construed as a warranty or commitment by PTC. PTC assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION.

**Registered Trademarks of Parametric Technology Corporation or a Subsidiary**

Advanced Surface Design, CADD5, CADDShade, Computervision, Computervision Services, Electronic Product Definition, EPD, HARNESSDESIGN, Info\*Engine, InPart, MEDUSA, Optegra, Parametric Technology, Parametric Technology Corporation, Pro/ENGINEER, Pro/HELP, Pro/INTRALINK, Pro/MECHANICA, Pro/TOOLKIT, PTC, PT/Products, Windchill, InPart logo, and PTC logo.

**Trademarks of Parametric Technology Corporation or a Subsidiary**

3DPAINT, Associative Topology Bus, Behavioral Modeler, BOMBOT, CDRS, CounterPart, CV, CVact, CVaec, CVdesign, CV-DORS, CVMAC, CVNC, CVToolmaker, DesignSuite, DIMENSION III, DIVISION, DVSAFEWORK, DVS, e-Series, EDE, e/ENGINEER, Electrical Design Entry, Expert Machinist, Expert Toolmaker, Flexible Engineering, *i*-Series, ICEM, Import Data Doctor, Information for Innovation, ISSM, MEDEA, ModelCHECK, NC Builder, Nitidus, PARTBOT, PartSpeak, Pro/ANIMATE, Pro/ASSEMBLY, Pro/CABLING, Pro/CASTING, Pro/CDT, Pro/CMM, Pro/COMPOSITE, Pro/CONVERT, Pro/DATA for PDGS, Pro/DESIGNER, Pro/DESKTOP, Pro/DETAIL, Pro/DIAGRAM, Pro/DIEFACE, Pro/DRAW, Pro/ECAD, Pro/ENGINE, Pro/FEATURE, Pro/FEM-POST, Pro/FLY-THROUGH, Pro/HARNESS-MFG, Pro/INTERFACE for CADD5, Pro/INTERFACE for CATIA, Pro/LANGUAGE, Pro/LEGACY, Pro/LIBRARYACCESS, Pro/MESH, Pro/Model.View, Pro/MOLDESIGN, Pro/NC-ADVANCED, Pro/NC-CHECK, Pro/NC-MILL, Pro/NC-SHEETMETAL, Pro/NC-TURN, Pro/NC-WEDM, Pro/NC-Wire EDM, Pro/NCPOST, Pro/NETWORK ANIMATOR, Pro/NOTEBOOK, Pro/PDM, Pro/PHOTORENDER, Pro/PHOTORENDER TEXTURE LIBRARY, Pro/PIPING, Pro/PLASTIC ADVISOR, Pro/PLOT, Pro/POWER DESIGN, Pro/PROCESS, Pro/REPORT, Pro/REVIEW, Pro/SCAN-TOOLS, Pro/SHEETMETAL, Pro/SURFACE, Pro/VERIFY, Pro/Web.Link, Pro/Web.Publish, Pro/WELDING, Product Structure Navigator, PTC *i*-Series, Shaping Innovation, Shrinkwrap, The Product Development Company, Virtual Design Environment, Windchill e-Series, CV-Computervision logo, DIVISION logo, and ICEM logo.

**Third-Party Trademarks**

Oracle is a registered trademark of Oracle Corporation. Windows and Windows NT are registered trademarks of Microsoft Corporation. Java and all Java based marks are trademarks or registered trademarks of Sun Microsystems, Inc. CATIA is a registered trademark of Dassault Systems. PDGS is a registered trademark of Ford Motor Company. SAP and R/3 are registered trademarks of SAP AG Germany. FLEX/m is a registered trademark of GLOBEtrouter Software, Inc. VisTools library is copyrighted software of Visual Kinematics, Inc. (VKI) containing confidential trade secret information belonging to VKI. HOOPS graphics system is a proprietary software product of, and copyrighted by, Tech Soft America, Inc. All other brand or product names are trademarks or registered trademarks of their respective holders.

---

---

**UNITED STATES GOVERNMENT RESTRICTED RIGHTS LEGEND**

This document and the software described herein are Commercial Computer Documentation and Software, pursuant to FAR 12.212(a)-(b) or DFARS 227.7202-1(a) and 227.7202-3(a), and are provided to the Government under a limited commercial license only. For procurements predating the above clauses, use, duplication, or disclosure by the Government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or Commercial Computer Software-Restricted Rights at FAR 52.227-19, as applicable.

**Parametric Technology Corporation, 140 Kendrick Street, Needham, MA 02494-2714**

**8 January 2001**

---



---

# Table of Contents

---

## Preface

Related Documents .....	xiii
Book Conventions .....	xiv
Online User Documentation .....	xv
Printing Documentation .....	xv
Resources and Services .....	xvi
Documentation Comments .....	xvi

## Introducing Vault

Describing Vault .....	1-2
Making Use of Vault .....	1-3

## Accessing Vault

Connecting to Vault .....	2-2
Access Authority .....	2-2
Accessing Vault .....	2-3
Using Common Login .....	2-4
Signing On from the Command Line .....	2-4
Changing Your Sign On .....	2-5
Leaving Vault .....	2-5

---

Using Basic Vault Transactions _____	2-6
Storing a Local Object in Vault _____	2-6
Changing the Default Local Directory _____	2-7
Reserving a File Name in Vault _____	2-8
Checking Out and Checking In an Object _____	2-8
Checking Out an Object _____	2-9
Checking In an Object _____	2-11
Duplicating a File _____	2-11
Sending and Reading Messages _____	2-12
Sending a Message _____	2-12
Reading a Message _____	2-14
Participating in a Review _____	2-16
Summarizing the Release/Revision Control Cycle _____	2-16
Responding to a Review Request _____	2-18
Learning Review Results _____	2-19
Initiating a Review _____	2-19
Setting Up a Review _____	2-19
Using the Distributed Vault _____	2-20
Registering an Object _____	2-20
Distributing an Object _____	2-21

## Understanding Vault Concepts

Defining Vault Objects _____	3-2
Part _____	3-2
CADDs Part _____	3-2
Custom Part _____	3-2
File Set _____	3-2
Binder _____	3-3
Vault Object Comparison _____	3-4
Naming Conventions in Vault _____	3-5

---

Defining Vault Object Properties _____	3-6
Object name _____	3-6
Revision _____	3-7
Status Level _____	3-8
Classification _____	3-9
Project _____	3-9
Project ID _____	3-10
Part Name _____	3-10
Local Rule _____	3-10
Password _____	3-10
Storage Pool _____	3-11
Vault Attributes _____	3-11
User-Defined Attributes _____	3-12
Attribute File _____	3-12
Viewing Object Properties _____	3-14
Changing Object Properties _____	3-15
Changing the Revision _____	3-16
Changing the Status Level _____	3-17
Changing the Classification _____	3-17
Changing the Password _____	3-18
Moving a File to a Different Storage Pool _____	3-18
Changing Properties of Grouped Objects _____	3-19

## Working with Objects

Storing an Object in Vault _____	4-2
Storing a Local File _____	4-2
Reserving a File Name in Vault _____	4-3
Storing a Local Directory _____	4-3
Storing a Part _____	4-4
Storing a CADD Part _____	4-4
Storing a Custom Part _____	4-6
Adding a Local File to a Stored Part _____	4-6

---

Checking an Object out of Vault _____	4-8
Checking Out a CADDs Part _____	4-9
Checking Out a File Set _____	4-9
Checking Out a Binder _____	4-9
Checking Out Partial Contents of an Object _____	4-9
Naming a Checked Out Object _____	4-9
UNIX Names of Checked Out Objects _____	4-11
Checking an Object Into Vault _____	4-13
Checking In a Part _____	4-13
Checking In a File Set _____	4-14
Checking In a Binder _____	4-14
Duplicating a Stored Object _____	4-15
Duplicating a Checked Out Object _____	4-15
Naming the New Object _____	4-16
Examples of Names of Copies _____	4-16
Marking an Object for Deletion or Archive _____	4-17
Marking an Object _____	4-17
Marking a Password-protected Object _____	4-18
Marking an Incomplete Object _____	4-19
Unmarking an Object _____	4-19

## Using File Sets

Creating a File Set _____	5-2
Creating and Naming a File Set _____	5-2
Adding a Member to a File Set _____	5-3
Recursive File Sets _____	5-3
Including Objects in Multiple File Sets _____	5-4
Adding a Reserved File Name to a File Set _____	5-5
Eliminating a Member from a File Set _____	5-6
Removing a Member from a File Set _____	5-6
Deleting All Members of a File Set _____	5-6

---

Removing a File Set or a File Set Name _____	5-8
Working with a File Set _____	5-9
Checking Out and Checking In a File Set _____	5-9
Null (or Empty) File Sets _____	5-9

## Using Binders

What Are Binders? _____	6-2
What Are Relationships? _____	6-3
Object Types and Attributes _____	6-4
Example of Object Type Entries _____	6-4
Attributes on Binders and Relations _____	6-5
Example of Attribute Group and Attribute Entries _____	6-5
Attributes on a Relation _____	6-6
Mandatory Attribute Groups and Attributes _____	6-7
Example: Mandatory Attribute Groups _____	6-8
Example: Optional Attribute Groups _____	6-9
Multivalued Attributes _____	6-9
Types of Relationships _____	6-11
Association _____	6-11
Characteristics of Associated Members _____	6-11
Example _____	6-11
Aggregation _____	6-12
Characteristics of Aggregated Members _____	6-12
Example 1 _____	6-12
Example 2 _____	6-13
Operations on Binders _____	6-14
Sign Out _____	6-14
Review _____	6-14
Overview of the Binder Process _____	6-15
Phase 1: Binder Setup _____	6-15
Phase 2: Working with Binders _____	6-15

---

Phase 1: Binder Setup _____	6-17
Defining an Object Type _____	6-18
Defining an Attribute Group Name _____	6-19
Defining an Attribute _____	6-21
Selecting the Attribute Data Type _____	6-23
Creating an Enumeration _____	6-23
Adding Values to an Enumeration List _____	6-24
Removing a Value from an Enumerated List _____	6-24
Phase 2: Working with Binders _____	6-25
Creating a Binder _____	6-25
Creating Class Attributes _____	6-27
Adding a Member to a Binder _____	6-29
Editing Attributes _____	6-31
Creating Instance Attributes _____	6-32
Viewing a Binder's Hierarchy _____	6-33
Removing a Member from a Binder _____	6-34
Checking a Binder Out _____	6-35
Checking a Binder In _____	6-36
Deleting a Binder _____	6-37

## Customizing the Interface

Edmgui Resource File _____	A-2
User Settings _____	A-2
System Administrator Settings _____	A-2
Calling Other Programs from the Interface _____	A-3
Preferences Menu _____	A-4
Dialog Box Customization _____	A-5
Font Customization _____	A-6

---

# Using CADD5 Commands to Access the Vault Database

Introduction to CADD5 5i and Vault _____	B-2
Starting the DBMS Process _____	B-2
Signing On to Vault from Within CADD5 5i _____	B-2
CADD5 Command Description Format _____	B-3
Description _____	B-3
Valid Modes _____	B-3
Format _____	B-3
Modifiers _____	B-3
Examples _____	B-4
Vault Commands in UNIX and CADD5 5i _____	B-4
GET PART _____	B-8
Valid Modes _____	B-8
Format _____	B-8
Modifiers _____	B-8
Example GET PART WING _____	B-10
Example GET PART WING READ A _____	B-10
PUT PART _____	B-11
Valid Modes _____	B-11
Format _____	B-12
Modifiers _____	B-12
Example PUT PART TEST.PART (CR) _____	B-13
UPDATE PART _____	B-14
Valid Modes _____	B-14
Format _____	B-14
Modifiers _____	B-14
Example UPDATE PART TEST.PART _____	B-15
GET FILE _____	B-16
Valid Modes _____	B-16
Format _____	B-16
Modifiers _____	B-16
Example GET FILE AA.HELP _____	B-17

---

PUT FILE _____	B-18
Valid Modes _____	B-18
Format _____	B-18
Modifiers _____	B-18
Example PUT FILE HELP NEW PUBLIC SINGLE _____	B-20
Example PUT FILE AA.HELP NEW PUBLIC _____	B-20
GET FILESET _____	B-21
Valid Modes _____	B-21
Format _____	B-21
Modifiers _____	B-21
Example GET FILESET TESTFS3 READ AB _____	B-22
Example GET FILESET TESTFS3 _____	B-22
PUT FILESET _____	B-23
Valid Modes _____	B-23
Format _____	B-23
Modifiers _____	B-23
Example PUT FILESET TESTFS3 _____	B-24
Quick Reference for Vault Commands in CADD5 5i _____	B-25
PUT PART _____	B-25
GET PART _____	B-25
UPDATE PART _____	B-25
GET FILE _____	B-25
PUT FILE _____	B-26
GET FILESET _____	B-26
PUT FILESET _____	B-26

## Corresponding Commands

Command Line and GUI Functionality _____	C-2
--	-----

## Glossary

---

# Preface

---

The *Vault End User Guide* provides information about Optegra enterprise data management solutions, such as Binders.

This guide describes the following:

- Storing objects in Vault
- Working with objects in Vault
- Customizing your Vault
- Corresponding menu options available on Vault with functionality provided by the command line

## Related Documents

The following documents may be helpful as you use *Vault End User Guide*:

- *Vault Manager Guide*

## Book Conventions

The following table illustrates and explains conventions used in writing about Optegra applications.

Convention	Example	Explanation
EPD_HOME	cd \$EPD_HOME/install (UNIX)  cd %EPD_HOME%\install (Windows)	Represents the default path where the current version of the product is installed.
Menu selections	Vault > Check Out > Lock	Indicates a command that you can choose from a menu.
Command buttons and options	Mandatory check box, Add button, Description text box	Names selectable items from dialog boxes: options, buttons, toggles, text boxes, and switches.
User input and code	Wheel_Assy_details  -xvf /dev/rst0  Enter command> <b>plot_config</b>	Enter the text in a text box or on a command line.  Where system output and user input are mixed, user input is in bold.
System output	CT_struct.aename	Indicates system responses.
Parameter and variable names	tar -cvf /dev/rst0 filename	Supply an appropriate substitute for each parameter or variable; for example, replace <b>filename</b> with an actual file name.
Commands and keywords	The ciaddobj command creates an instance of a binder.	Shows command syntax.
Text string	"SRFGROUPA" or 'SRFGROUPA'	Shows text strings. Enclose text strings with single or double quotation marks.
Integer	n	Supply an integer for <i>n</i> .
Real number	x	Supply a real number for <i>x</i> .
#	# mkdir /cdrom	Indicates the root (superuser) prompt on command lines.
%	% rlogin remote_system_name -l root	Indicates the C shell prompt on command lines.
\$	\$ rlogin remote_system_name -l root	Indicates the Bourne shell prompt on command lines.
>	> copy filename	Indicates the MS-DOS prompt on command lines.
Keystrokes	Return or Control-g	Indicates the keys to press on a keyboard.

---

## Online User Documentation

Online documentation for each Optegra book is provided in HTML if the documentation CD-ROM is installed. You can view the online documentation from an HTML browser or from the HELP command.

You can also view the online documentation directly from the CD-ROM without installing it.

From an HTML Browser:

1. Navigate to the directory where the documents are installed. For example,  
\$EPD\_HOME/data/html/htmldoc/ (UNIX)  
%EPD\_HOME%\data\html\htmldoc\ (Windows NT)
2. Click `mainmenu.html`. A list of available Optegra documentation appears.
3. Click the book title you want to view.

From the HELP Command:

To view the online documentation for your specific application, click HELP. (Consult the documentation specific to your application for more information.)

From the Documentation CD-ROM:

1. Mount the documentation CD-ROM.
2. Point your browser to:  
CDROM\_mount\_point/htmldoc/mainmenu.html (UNIX)  
CDROM\_Drive:\htmldoc\mainmenu.html (Windows NT)

## Printing Documentation

A PDF (Portable Document Format) file is included on the CD-ROM for each online book. See the first page of each online book for the document number referenced in the PDF file name. Check with your system administrator if you need more information.

You must have Acrobat Reader installed to view and print PDF files.

The default documentation directories are:

- \$EPD\_HOME/data/html/pdf/doc\_number.pdf (UNIX)
- %EPD\_HOME%\data\html\pdf\doc\_number.pdf (Windows NT)

## Resources and Services

For resources and services to help you with PTC (Parametric Technology Corporation) software products, see the *PTC Customer Service Guide*. It includes instructions for using the World Wide Web or fax transmissions for customer support.

## Documentation Comments

PTC welcomes your suggestions and comments. You can send feedback in the following ways:

- Send comments electronically to [doc-webhelp@ptc.com](mailto:doc-webhelp@ptc.com).
- Fill out and mail the PTC Documentation Survey located in the *PTC Customer Service Guide*.

# Introducing Vault

---

This chapter provides a brief overview of Vault. It lists the objects you can store in Vault and their associated properties. It also presents the advantages of using Vault to store your work.

- Describing Vault
- Making Use of Vault

# Describing Vault

Vault is a data management tool that helps automate the entire process of product development. With Vault, you can centralize control of design databases and ensure that there is no unauthorized access to product data. The purpose of Vault is to make it easier for you to manage the process of designing and manufacturing a product.

Vault can contain single files or files grouped in the following categories:

- Part
- File set
- Binder

Objects stored in Vault have required and optional information, called properties, associated with them. You provide this information when you store the object:

- Object Name
- Revision
- Classification
- Status Level
- System Type
- Description
- Group Technology Code
- Password
- Part Name
- Local Rule
- Attribute File
- User-Defined Type
- Part Number
- Project ID

# Making Use of Vault

Vault performs the following functions:

- Centralizes storage for many kinds of files
- Controls data access in a distributed environment
- Limits access to the database to authorized users
- Ensures that only one person at a time can modify a file
- Allows designers and engineers users to transfer individual files and parts as well as sets of files and parts to and from Vault database
- Provides a method for relating a group of files and/or parts so that you can treat them as a single unit
- Differentiates among the following categories of files:
  - File revisions
  - Files belonging to different projects
  - In-work versus released files
  - Private files
  - Public files
- Provides a mechanism for sending and reading messages



# Accessing Vault

---

This chapter provides an overview of Vault, along with basic instructions for signing onto and leaving Vault, storing objects, checking stored objects out of and into Vault, reserving file names, and changing information associated with stored objects.

The following sections describe these topics.

- Connecting to Vault
- Using Basic Vault Transactions
- Sending and Reading Messages
- Participating in a Review
- Using the Distributed Vault

Please note: Although the procedures presented here all indicate files as the selected object, in most cases you can substitute other objects with little modification. Each procedure notes object limitations, when applicable.

## Connecting to Vault

Before you can access Vault, you must ask your Vault Administrator for the following:

- User ID (up to 12 characters)
- User password (up to 8 characters)

Please note: Do not use Kanji characters within a password.

- Vault server name, which is also the Vault name.

Please note: This information is not required if Common Login is enabled. For more information see “Using Common Login” on page 2-4.

## Access Authority

Authority associated with your Vault User ID determines which Vault files and parts you can access.

The Vault Administrator assigned you public read and write authority during creation of your Vault User ID. Your public authorities determine your access to public files and parts.

You can have different authorities for each assigned project. If you attempt to perform an action for which you do not have the authority, Vault displays a message and does not allow the action.

Your Vault User ID also contains authority information which allows or prohibits customization of your Vault sign-on procedure and the appearance of the Vault graphical user interface itself. For details, see Appendix A, “Customizing the Interface.”

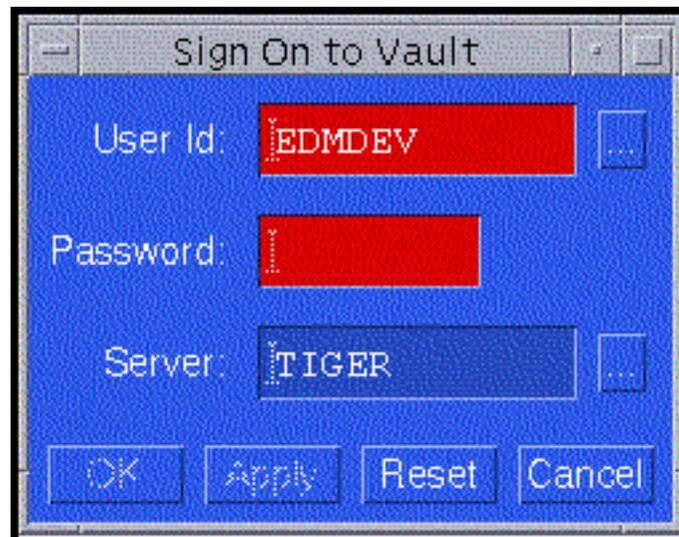
Please note: Since the Vault graphical user interface is customizable, your installation’s Vault appearance may differ from the figures in this guide.

## Accessing Vault

The first time you invoke the Graphical User Interface to access Vault, or if you invoke it with Common Login disabled (detailed in “Using Common Login” on page 2-4), follow this procedure:

1. At the system prompt, type `edmgui`. The Sign On to Vault dialog box appears.

**Figure 2-1 Sign On to Vault Dialog Box**



2. Type your User ID (or access it with the Editor button). Click the Password box. The User box turns from red to blue.
3. Type your user password. Click in the Server box. The Password box turns from red to blue.
4. Type the name of the Vault you want to access, if it is not already displayed.
5. Click OK. A confirmation dialog box appears and displays the number of messages you have.

Please note: Red text boxes indicate either required or invalid data. You must fill in these boxes correctly and click in another box (or use the Tab key) to activate them before you can choose OK or Apply. Activated boxes turn blue.

6. Click OK in the confirmation dialog box. The Vault Files browser is displayed.

Please note: If you want to read your messages, you can access Messages > Read from the menu bar. You can send messages with the Messages > Send option. For details see “Sending and Reading Messages” on page 2-12.

## Using Common Login

The first time you sign on to Vault, enter the user ID and password.

Please note: If you later sign on as a different user, Vault replaces the stored values with the new ones.

For more information, refer to *Installing Optegra Applications*.

## Signing On from the Command Line

To sign on to Vault from the command line, type:

```
cisignon
```

If Common Login is enabled, the same conditions apply as login from the Graphical User Interface:

- The first time you login with Common Login enabled, you must enter a User ID and password. If you do not enter this information when you type `cisignon`, the system prompts you for the information. Your User ID and password are automatically stored in your local common login repository.

Please note: Do not use Kanji characters within a password.

- Subsequent logins with Common Login enabled that include a User ID and password overwrite the information stored in your local common login repository.
- Once Vault creates your local common login repository, subsequent logins with Common Login enabled that do not include a User ID and password place you directly in Vault using the most recent values for this information entered in that file.

Please note: If you type `edmgui -nclogin` on the command line, the Sign On to Vault dialog box appears. You can then proceed as described in the section “Accessing Vault” on page 2-3.

## Changing Your Sign On

You can sign on to another Vault or use another User ID provided you have the necessary access permissions.

1. From the menu bar, choose File> Sign On. The Sign On to Vault dialog box appears.
2. Choose the User ID or the Server you want, or both.
3. Type your Password and click in another box.
4. Click OK.

## Leaving Vault

When you have concluded your work in Vault, you need to exit the from the graphical user interface.

1. From the menu bar, choose File > Exit. The Edmgui Information dialog box appears.
2. Click Yes (this is the only option).

To sign off of Vault, substitute Sign Off for Exit in step 1.

When you Sign Off, you leave the Vault in which you are working, but you remain signed on to your User ID. You can still view local objects, but you can not perform any Vault activities on them. You can, however sign on to another Vault from File > Sign On on the menu bar.

When you Exit, you leave the graphical user interface entirely and must sign on from the command line or `edmgui` if you want to re-access Vault. Any preferences you have set during your session revert to the original settings, unless you saved them in your `$HOME/Edmgui` file. See Appendix A, “Customizing the Interface,” for information about the `.Edmgui` file and setting Vault preferences.

Please note: Sign off instead of exiting when the Vault in which you are working notifies you that it going down. This enables you to maintain any preference customization you performed during your session for use when you sign back on when the Vault becomes available.

## Using Basic Vault Transactions

Vault allows you to perform the following basic transactions:

- Store a file in Vault from your local storage area
- Reserve a file name
- Check out and check in a file
- Duplicate a file already stored in Vault
- Set and change file properties (information about the file, not the file contents)

These tasks and their associated options appear in the table titled Table 2-1, “Vault Menu Commands” on page 2-8.

Please note: This chapter provides an introduction to basic Vault functionality. For detailed information, see Chapter 4, “Working with Objects.”

### Storing a Local Object in Vault

To make a local object available to other Vault users, you must move it from your local storage area to Vault. You can store the following local objects in Vault:

- File
- Part
- Directory of files

Once an object is stored in Vault, Vault manages it according to the authority schemes specified in the user/project authority schemes associated with your User ID.

When you store an object, Vault deletes it from your local storage area unless you direct otherwise. You can indicate whether or not to leave an object signed out to your local area when you store it by clicking on the Leave Signed Out button on the Store in Vault dialog box. The default is off.

You can modify the following procedure to store other kinds of objects in Vault by indicating the type of object you want to store from the View > Local menu.

1. From the menu bar, choose View > Local > Files. This places you in the Enterprise Data Manager - Local Files browser.
2. Choose an item in the browser

3. From the menu bar, choose Vault > Store. The Store in Vault dialog box appears.
4. Fill in the information required. If you change the name in the Object Name box, Vault stores the object using the new name.
5. Click OK (or Apply if you want to keep the dialog box open).

Please note: OK and Apply serve different purposes. When you click OK, Vault acts on your instructions and closes the dialog box. When you click Apply, Vault acts on your instructions but leaves the dialog box open so you can use it to define other objects without reaccessing it from the menu bar. You can keep multiple dialog boxes active. The procedures in this guide direct you to click OK, unless Apply is required.

When you store an object in Vault for the first time, if you don't specify information about some of the object's properties, Vault provides default values.

After you store an object, you can access, view or modify its properties through the Edit item on the menu bar. You can associate combinations of stored files and parts to create file sets and binders, which is described later in this guide.

Chapter 3, "Understanding Vault Concepts," describes Vault objects and their associations. Binders (containers for logically related objects) are detailed in Chapter 6, "Using Binders."

## Changing the Default Local Directory

The default local directory into which Vault stores your object is specified in Preferences > Current Settings, which you can access from the menu bar. You can override this default for your current session.

1. Choose View > Local > Files to go to the Local Files browser.
2. Choose Edit > Selection Criteria. The Edit Local Selection Criteria dialog box appears.
3. Type the new default local directory in the Base Directory dialog box.
4. Click OK. The Local Files browser now displays the files in the directory you specified.

Please note: For details on customizing the user interface see Appendix A, "Customizing the Interface."

## Reserving a File Name in Vault

You can reserve a file name in Vault as a placeholder for future use. When you do this, you can also specify properties such as Classification, Project ID, and File Status Level for association with the file name. Specify physical data when you store the actual file.

Please note: Vault allows you to reserve a file name, only. You can not reserve any other object name in Vault.

1. From the menu bar, choose View > Local > Files.
2. From the menu bar, choose Vault > Reserve. The Reserve a Name dialog box appears.
3. Fill in the boxes on this dialog box.
4. Click OK.

You can work with the reserved file name as though it were an actual file stored in Vault. When you are later ready to associate a local file with this reserved file name, use the Store procedure.

Please note: You cannot change the properties of a reserved file name when you associate a file with it in Vault. For details on changing object properties, see Chapter 3, “Understanding Vault Concepts.”

You can remove a reserved file name if you no longer need it.

1. Choose the object from the Vault Files browser.
2. From the menu bar, choose Vault > Unlock. This resets the lock.
3. To remove the file name, again choose Vault > Unlock.
4. Click OK.

## Checking Out and Checking In an Object

You can check a stored object out of Vault and modify it. You can then check it back in and indicate its state in Vault. The following list describes the different states of objects in Vault.

**Table 2-1 Vault Menu Commands**

Command	Action	Results	State of Object
Store	Stores a file, directory, or a part in Vault for the first time.	Object is stored in Vault. Local version is deleted.	You can modify the object, unless otherwise specified in the Attribute file.
Check Out > Lock	Checks out file, part, list, file set, or binder.	Object is transferred to your local area.	You can modify the checked out object. Other users can not check it out

**Table 2-1 Vault Menu Commands**

<b>Command</b>	<b>Action</b>	<b>Results</b>	<b>State of Object</b>
Check Out > Do Not Lock	Checks out file, part, file set, or binder.	Object is transferred to your local area.	You can modify the checked -out object. Other users can still check it out.
Check Out > Lock Only	Checks out file, part, file set, or binder.	Object is not transferred to your local area.	You cannot modify the object unless you unlock it and check it out (locking it again).
Reserve (Available from View > Local Files, Only)	Reserves a unique file name in Vault for future use.	Only the file name and associated properties are created. There is no file version associated with the file name until you store the file in Vault.	You cannot modify the file name's associated properties unless you store it in Vault first.
Check In > Unlock	Transfers the local version back to Vault.	Object is no longer signed out to you. Local version is deleted.	Vault object is changed. You can modify the object.
Unlock	Does not transfer the local version back to Vault. The Vault version remains unchanged, although changes may exist on the local version.	Object is no longer signed out to you. Local version is not deleted.	Vault version is not changed. Local version is not deleted. You cannot modify the object unless you lock it.
Duplicate	Duplicates the indicated file, or a part stored in Vault and modifies properties you specify.	No local version unless you check out the one stored in Vault.	Object stored in Vault. You can modify the object even if the original was locked.
Check In > Keep Lock	Checks the local version back into Vault.	Object remains signed out to you.	Vault version is updated. Local version is not deleted. You can modify the object.

## Checking Out an Object

You must check an object out before you can modify it. You can check a binder, a part, a file set, or an assembly out of Vault.

When you check an object out of Vault, you create a copy of the object in a local directory. At this time, indicate whether or not Vault should permit access to the copy which remains by either locking the object or leaving it unlocked.

The local directory into which Vault copies your checked-out object is specified in your Preferences, which you can access from the menu bar. You can override this default by specifying a Local Directory Name on the Checkout dialog box.

Please note: To check out a set of individual files initially stored as a directory, each file must be defined as a member of a file set. Chapter 5, “Using File Sets,” describes grouping objects into a file set.

1. From the menu bar, choose View > Vault > Files.
2. From the Vault Files browser, choose the specific file you want to check out.
3. From the menu bar, choose Vault > Check Out.
4. Choose the option to describe the state in which you wish to leave the Vault copy of file. (Table 2-1, “Vault Menu Commands” on page 2-8 describes these choices.)

If you check out a file set that includes a part and intend to add a file to the part or delete a file from the part, do not designate Lock or Lock Only.

5. When the Checkout dialog box appears, fill in the boxes appropriately.

If you do not specify a Local Directory Name, Vault checks your file out to the default directory.

6. Click OK.

You cannot check out and lock an object that is:

- Signed out to you or to another user
- Marked for archive
- Marked for deletion
- Locked by the Release/Revision Control facility

The results of an attempt to check out an object containing unavailable members (members already checked out or locked, for example) differ, depending on the selected type of object.

For parts and file sets, none of the part or file set members are checked out.

## Checking In an Object

Check an object into Vault when you are finished modifying the checked out copy, to update the Vault copy, or to change the state of the copy still in Vault.

Please note: Vault automatically checks an object in with the same name it had when you checked it out. This is true even if you renamed the object when you checked it out. However, if you move the file to a different local storage area, Vault loses track of it and you must either move it back to the area in which you checked it out so Vault can locate it or store it like a new file, rather than check it in.

1. From the menu bar, choose View > Vault > Files.
2. From the Vault Files browser, choose the file you want to check in.
3. From the menu bar, choose Vault > Check In.
4. Choose the option to describe the state of the Vault file when it is checked in. (Table 2-1, “Vault Menu Commands” on page 2-8 describes these options.)
5. Click OK.

## Duplicating a File

You can duplicate a file or file name that is stored in Vault. When you do this, you can also specify different properties for the new version.

1. From the menu bar, choose View > Vault > Files.
2. From Vault on the menu bar, choose Duplicate. The Save As dialog box appears.
3. Type the new name and the values for any properties you want to change.
4. Click OK.

## Sending and Reading Messages

The Vault message facility allows two types of Vault messages:

- Notification messages, which does not require a response
- Review messages, which generally requires a response

The Release/Revision Control facility also sends out Notification messages to call attention to two circumstances:

- The initiation of a file review
- The existence of an issue resulting from a file review

Messages sent by the Release/ Revision Control facility are detailed in the *Vault Manager Guide*.

### Sending a Message

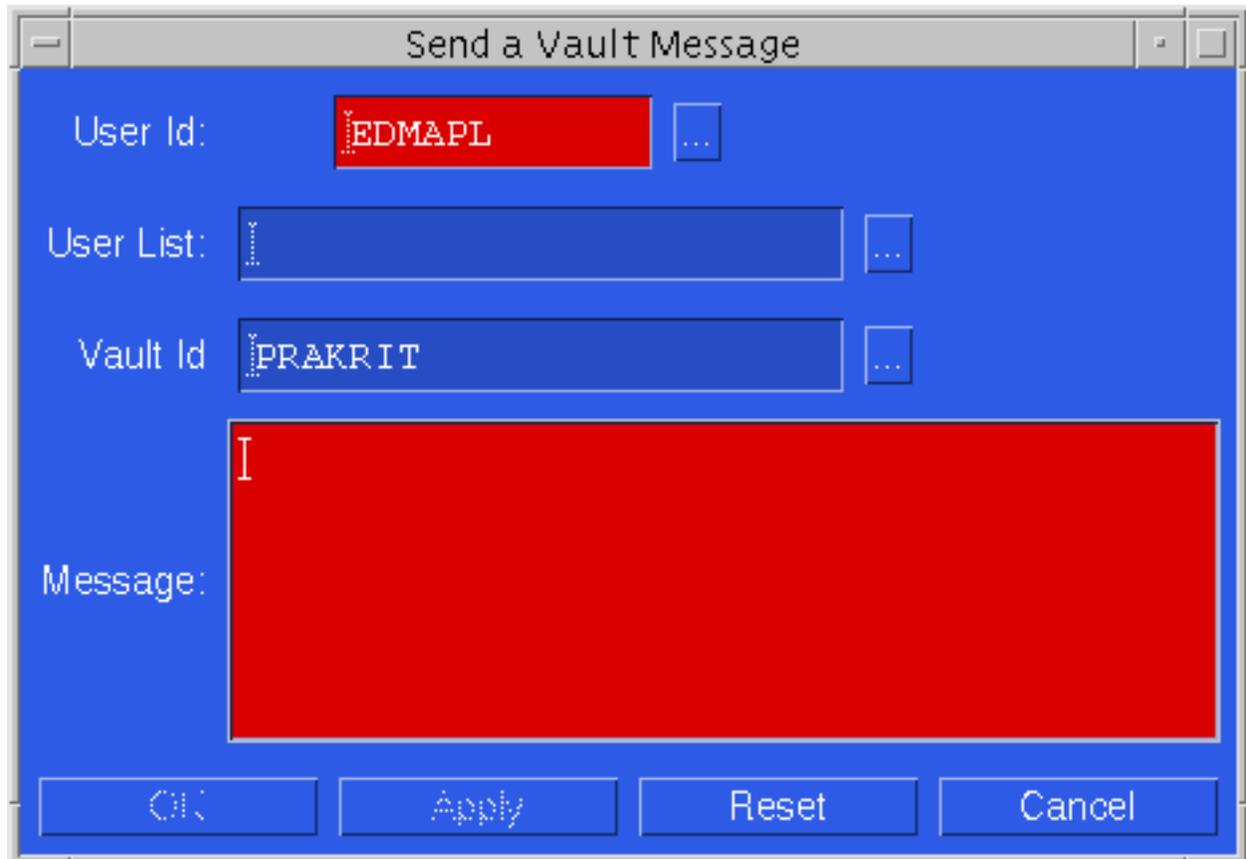
With one action, you can send a message to a variety of recipients:

- Vault user
- List of Vault users (called a User List)
- Vault user and a user list

Access the Messages option from the menu bar:

1. Choose Messages > Send. The Send a Vault Message dialog box appears.

**Figure 2-2 Send a Vault Message Dialog Box**



2. Enter the User ID, User List, Vault ID, and Message in the text boxes.
3. Click OK.

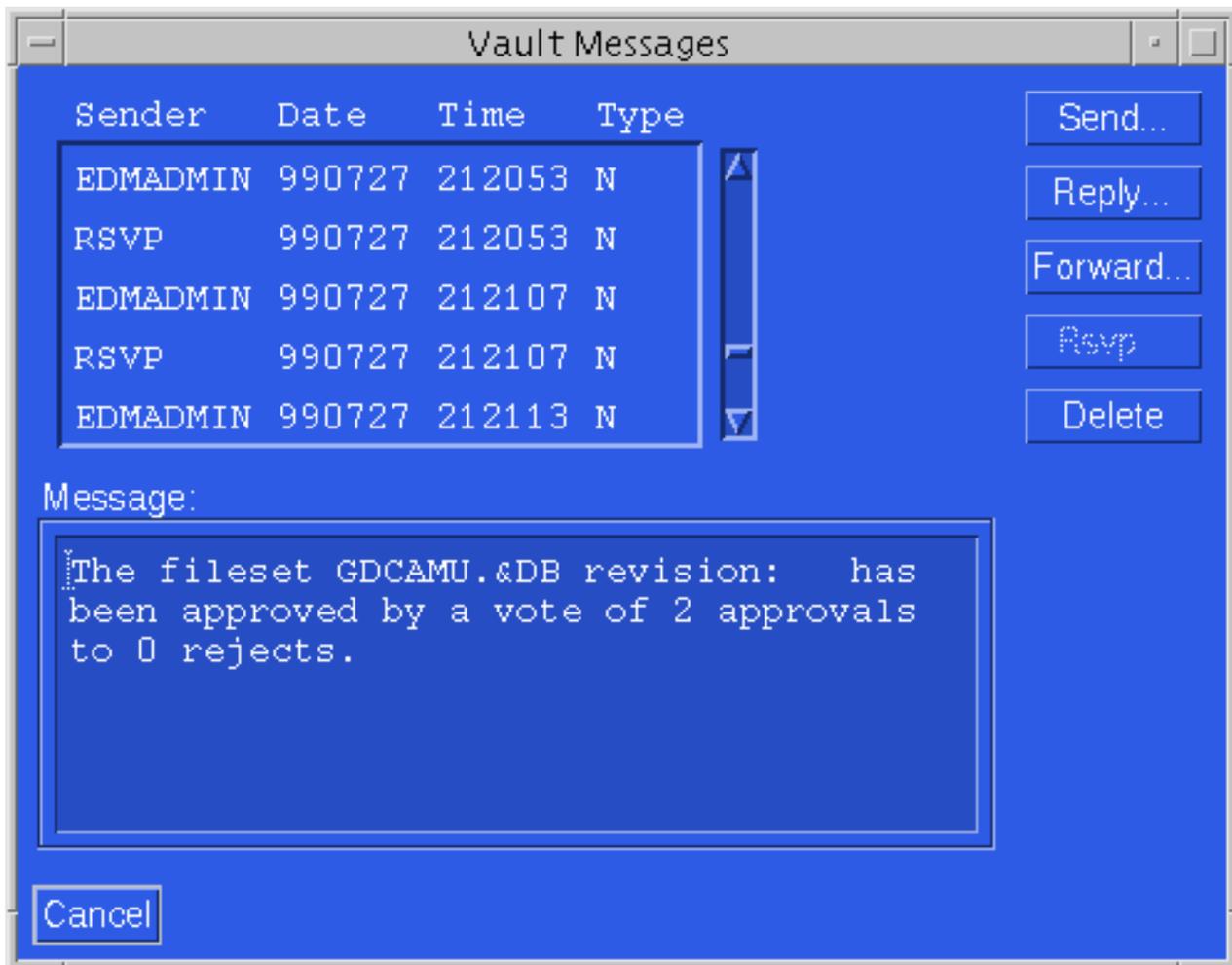
Please note: To keep copies of the messages you send, include your own Vault User ID on the specified user list or enter your own Vault User ID in the User ID dialog box.

## Reading a Message

The Status line at the bottom of the Vault browser notifies you if you have any messages. The number accounts for all messages, new and old. You can access both old and new messages in the following manner.

1. Choose Messages > Read. The Vault Messages dialog box appears.

**Figure 2-3 Vault Messages Dialog Box**



2. Click the message you want to read from the scrolled list. The full message appears in the Message text box.
3. When you are through, click Cancel.

Please note: From this dialog box, you can also Send a new message, Reply to the highlighted message, Forward the message to another user, or Delete the message.

There are two possible entries under the Type category in this dialog box:

- R indicates that the message is requesting a review of an object. The Release/Revision Control facility initiates a review request. For instructions on responding to this message type, see “Responding to a Review Request” on page 2-18.
- N indicates that the message is a notice. A notice can be a message from another Vault user or the results of a review.

## Participating in a Review

Vault allows authorized users to request and perform reviews of the following:

- Files
- Parts
- File sets
- Binders

Please note: For information on binders, see Chapter 6, “Using Binders.”

Before review, all files in either a part or a file set must have the same status code. You can not initiate a review on an individual file that belongs to a part; you must put the entire part into review.

If a status code is not associated with a review list, Vault automatically increments a file’s status when you request the review. It then notifies users on the post-approval list of the automatic status change.

## Summarizing the Release/Revision Control Cycle

A Vault object (file, part, or binder) evolves along a predefined path from the time it is stored until it is reviewed and released. Vault indicates the object’s state of completion by associating a sequence of status levels with it.

Each set of status levels composes a revision. The revision code of a Vault object distinguishes the evolutionary stage of that object.

While a single revision in the evolution of an object may be referred to as a design cycle, the set of all design cycles, (as represented by the set of all revisions of the object), can be thought of as the object’s life cycle.

When a stage of work on a file completes, a review cycle commences to determine if the file is ready for the next status in the designated authority scheme.

Please note: Your Vault Administrator may have set up Vault to trigger email and messages to initiate and track a review process. Check with the Vault Administrator.

For example, at the High Flying Airplane Company (our sample site), drafting of all CARGO project part files has completed. Before changing the Status Level of

these files, a file review must take place. The sequence of events for a part file review follows:

1. The CARGO project leader executes the request for a review, specifying the part name to be reviewed.
2. Vault locks the part files so no one can sign them out for further modifications.
3. Vault sends a message to members of the Notification Subscription on the CARGO project (Beryl, Orville, Snoopy, and Wilbur). This message informs them that the review has been initiated. No response from them to this message is required.

Please note: Do not lock files before you start the review. Vault locks them during the review process. Reviewers can copy files into their local storage area if they need access (or execute the CADDs `get` command with the `read` option).

4. Vault sends a message to each review list member (Beryl, Orville, and Wilbur), in the order defined by the Vault Administrator or the project leader in the Notification Subscription. This message requests the recipient to review the specified files.

Please note: Reviewers may receive the review message at different times, depending on the Notification Subscription definitions.

- Sequential: Each reviewer must register approval or rejection before the next reviewer gets a review message.
  - Concurrent: All reviewers get the message at the same time and can respond in any order.
  - Mixed: A combination of sequential and concurrent.
5. Each reviewer inspects the file and registers approval or rejection, along with their comments, with the `RsvP` (respond to a review request) command.
  6. After the final reviewer enters a response, the file's Status Level either changes or remains the same, based on the review results. Vault interprets the results in one of three ways, depending on the review definition indicated by the Vault Administrator or project leader.
    - Unanimous: All reviewers must approve the file.
    - Majority: The majority of file reviewers must approve the file.
    - Number of rejects: The number of rejections must be less than or equal that specified at the inception of the review.

Please note: The Vault Administrator or project leader indicates the number of rejections which trigger review termination independently of the review approval scheme.

A file review continues until one of these events occurs:

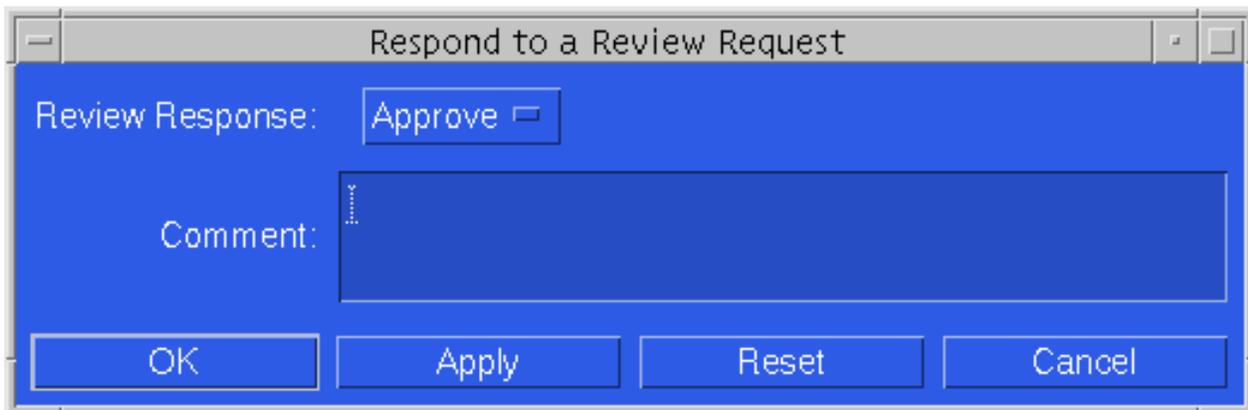
- All reviewers have responded to the review.
  - The maximum number of rejections specified for termination is reached.
7. Vault automatically notifies the file review initiator of the review results. If the review is successful, it results in a file Status Level change, and Vault notifies the voting reviewers, the post-approval list members, and the Notification List members. No response to this message is required.
  8. Vault unlocks the file, allowing check-out by users with the corresponding authority. (If the review results in a change in the file's Status Level from an in-work status to a released status, Vault increments the file's Revision, and initial in-work Status Level at the next check out.)

## Responding to a Review Request

You can respond to a review request.

1. Choose Messages > Read from the menu bar to read the requestor's comment.
2. Choose View > Vault from the menu bar and choose the type of object in review from the submenu that appears.
3. Either click Rsvp when you read the requesting message  
or  
Choose Review > Respond from the menu bar  
The Respond to a Review Request form appears.

**Figure 2-4 Respond to a Review Request Form**



The screenshot shows a dialog box titled "Respond to a Review Request". The dialog has a blue background. At the top, there is a label "Review Response:" followed by a dropdown menu with "Approve" selected. Below this is a label "Comment:" followed by a large, empty text input area. At the bottom of the dialog, there are four buttons: "OK", "Apply", "Reset", and "Cancel".

4. Click either Approve or Reject, typing your reason in the Comment box.

## Learning Review Results

When a review is complete, Vault sends a message describing the results of the review to the initiator, all users who voted on the review, and all members of the Notification List and post-approval list.

Review responses and comments are stored for future reference in an Oracle table. You can access the information using SQL\*Plus or by executing the `ciiqf` command at the command line, which invokes Vault's query facility IQF. For details on how to use IQF refer to the *Vault Interactive Query Facility Guide*.

If the file is approved by the required number of reviewers, Vault increments the file's status code and the file can be signed out for the next stage of work.

If the file is not approved, it remains at the same stage of work and its status code is not changed. Once the review process ends, the file is again accessible.

## Initiating a Review

You can initiate a review of an object through the Distributed Vault view if the Vault on which the object is stored is active.

This procedure for initiating a review is described in the *Vault Manager Guide*.

## Setting Up a Review

Your Vault Administrator may have set up Vault to trigger email and Vault messages to initiate and track a review process when a specified action occurs on a registered object. Check with the Vault Administrator for further information.

The procedure for establishing release/revision controls is described in the *Vault Manager Guide*.

## Using the Distributed Vault

The Distributed Vault allows you to access objects across multiple servers (Vaults) connected in a distributed, cooperative arrangement. This transparency allows you to access registered objects from any connected Vault without regard to location, as long as you have adequate privileges. Just as an object must be stored in the standalone Vault before you can act on it, an object must first be registered for transparent access in the distributed environment.

Vault transparency helps you locate and access the most suitable version of an object you want. The object's suitability depends on your authority on the Vault in which it is stored and the access defined when the object was registered.

The Distributed Vault browser only displays registered objects if you have access to the owning Vault. You must also know the object's name and type and satisfy the same conditions as a local user to access it. For example, you may see an object in the browser with a Registration Scope indicating higher accessibility than you are allowed. In this case, you can access the object at your authority level, rather than the object's registration level.

To access a file in the distributed view, you need not sign on to the Vault where the file resides. Sign on to one Vault and choose Distributed > Files or Distributed > Parts from View on the menu bar. This changes the browser's view to a display of the registered files or parts to which you have access. Any relevant review messages are routed to the Vault onto which you are signed.

You can work with objects in the Distributed Vault in the same way you would work with objects in the standalone Vault.

## Registering an Object

When you register an object, you make it available in the distributed view. You can only register an object that currently exists in Vault.

At registration, you indicate the object's degree of availability:

- If an object is registered for READ-ONLY access, any Vault user with adequate privileges can read it through the distributed view.
- If an object is registered for READ/WRITE access, any Vault user with adequate privileges can access it through the distributed view and perform actions which require write access, such as locking or updating the object.

Please note: An object registered with READ/WRITE access allows this privilege to all local Vault users.

If an object is not registered, only a user signed on to the Vault in which the object is stored can access the object.

The *Vault Manager Guide* details the registration process.

## Distributing an Object

Distributing an object sends a version of it to a different Vault using the Vault's Export/Import functionality. With proper permissions, you can distribute a file or part either on demand or by indicating an event that automatically triggers distribution.

Before you can designate an object for distribution, you must log into the Vault on which it is stored. The object must be registered so you can then select it from the Distributed Files or Parts browser for distribution.

There are six distribution types:

- Read Replica
- Write Replica
- Copy
- Move
- Update Original of a write replica
- Return Original of either a read or write replica

The *Vault Manager Guide* details these distribution types and the distribution process.



Objects stored in Vault have specific definitions and associations. The first three sections of this chapter describe Vault objects and properties. The last three sections detail changing the properties of Vault objects.

- Defining Vault Objects
- Naming Conventions in Vault
- Defining Vault Object Properties
- Viewing Object Properties
- Changing Object Properties
- Changing Properties of Grouped Objects

# Defining Vault Objects

Vault can contain single files or files grouped in the following categories:

- Part
- File set
- Binder

## Part

A part is a set of files recognized as a unit by an application such as CADD5 5i. A file can belong to only one part. Part members often share the same name prefix. Each file in a part, and the part itself, must share the same values for the properties Revision, Classification, and Status Level.

An operation performed on a part must affect the entire part, not an individual part member file. If an operation fails on a single part file, it fails on the entire part.

### CADD5 Part

A CADD5 part is one created by CADD5 5i software. A CADD5 part requires certain files in order to be valid CADD5 parts: there is always a `_pd` file and an `_fd` file.

### Custom Part

Your site may have custom parts produced by a CAD/CAM system other than CADD5 5i or by some other software. In some cases, these parts may consist of only one file. In other cases, part files may not share the full part name.

## File Set

A file set is a user-defined collection of parts, files, and/or other file sets. A file set has no Classification, Project ID, Revision, Attributes, or Status Level. The parts and files in one file set are not required to share the same Classification, Project ID, Revision, Attributes, or Status Level.

Some operations that manipulate files in Vault can operate on a file set as well as on individual files. Vault executes the operation for all or none of the file set members.

This Vault feature can save time when you want to perform the same operation on related files and also minimizes the possibility of leaving out important information.

A file set member can be a member of multiple file sets and can even be a member of the same file set multiple times, although recursive file sets are not allowed.

## Binder

A binder is a container for logically related objects. You can use a binder to collect parts, files, or other binders. Once a binder is created, you can access all of the objects in the binder with a single action.

A binder can have a Revision and Status Level different than its members' values for these properties. However, both the binder and its members must have the same Classification and Project Id.

You can use a binder to create the following collections:

- A work package containing parts and related documents
- A design review folder of drawing objects
- A group of contract deliverables such as drawings, reports, and documents

Chapter 6, "Using Binders," discusses binder concepts and functionality in greater detail.

## Vault Object Comparison

Vault object conventions differ. The following table compares the rules and requirements of the Part, File Set, and Binders.

**Table 3-1 Comparison of Vault Collections of Objects**

<b>Object</b>	<b>Members</b>	<b>Members' Required Shared Values</b>	<b>Grouped Members Shared or Owned by Object?</b>	<b>Multiple Copies of a Member Allowed?</b>
Part	Files only	Classification Project ID Revision Attributes Status Level	Owned	No
File Set	Files Parts Other file sets (multi-leveled hierarchies allowed)	None	Shared	Yes
Binder	Files Parts Other binders	Classification Project ID Other properties dependent on binder relationship defined	Dependent on binder relationship defined	Yes

## Naming Conventions in Vault

Each Vault file or part must have its own unique name (80 or fewer alphanumeric characters). This name can contain any ASCII character found on your keyboard. This character set includes uppercase and lowercase letters, numbers, and special characters, although you can not include blank spaces in Vault names. Vault also ignores blank spaces before and after the name.

Vault maps an object's local name to uppercase when you store or check in a file, unless the `EDMCASE` environment variable in your `EDM.DEFAULTS` file is set to `LOWER` or `MIXED`.

When you check an object out of Vault, the local name follows the naming format, character set, and syntax of the operating system on which you are working. Because UNIX is case-sensitive, the name of a checked-out object appears as it was before it was stored, regardless of any changes Vault made in accordance with the `EDMCASE` environment variable setting.

Please note: CADDs files are always stored in uppercase in Vault.

## Defining Vault Object Properties

Objects stored in Vault have required and optional information, called properties, associated with them. You provide this information when you store the object.

Boxes for object properties appear on the Store In Vault dialog box when you first store the file. Any default values display in the Store In Vault dialog box. The following sections detail the object properties.

- Object Name
- Revision
- Classification
- Status Level
- System Type
- Description
- Group Technology Code
- Password
- Part Name
- Local Rule
- Attribute File
- User-Defined Type
- Part Number
- Project ID

Files stored in Vault may have some or all of these properties defined. Other objects allow association with only a subset of these properties.

If you are storing a file with the Classification of PROJECT, you must also indicate a Project ID. For details see “Project ID” on page 3-10.

Please note: If you do not indicate the Object Name, Revision, Status Level or Classification when you store a file, Vault determines default values.

### Object name

When you store an object, if you do not specify a name, Vault gives it the local name, (which you chose in the Local Files browser). This name displays in the Object Name box in the Store In Vault dialog box.

## Revision

A Vault object (file, part, or binder) evolves along a predefined path from the time it is stored until it is reviewed and released. Vault indicates the object's state of completion by associating a sequence of status levels with it.

Each set of status levels composes a revision. The revision code of a Vault object distinguishes the evolutionary stage of that object.

While a single revision in the evolution of an object may be referred to as a design cycle, the set of all design cycles, (as represented by the set of all revisions of the object), can be thought of as the object's life cycle.

When you store a file in Vault for the first time you can specify a Revision. When you do not specify a Revision, Vault automatically assigns the initial revision code in the revision code sequence (defined by the Vault Administrator) for the object, based on the Classification.

Please note: You can store a file indicating a higher Revision, although this may cause problems. You may find this functionality helpful, however, if you wish to Check In an object with the same name as one that already exists in Vault with an in-work Status Level. To work around this, Store your object with the same name, but change the Status Level to indicate that the object is released and specify a different Revision than that of the original.

When you try to access an object (using Check Out, Lock Only) on which the life cycle has completed, Vault indicates that the current object revision is checked out.

When you select Check In, Keep Lock, Vault copy updates, and the file name entry for the next revision level remains in your local files directory.

When a new life cycle begins on a released revision of a Vault file or part, Vault automatically prepares the object:

1. It creates a placeholder for the released object in Vault until the object is checked in or updated.
2. It assigns the copy the next revision code in a previously defined sequence.
3. It assigns the copy the initial in-work status code.

The following table details the effect of duplicating an object in Vault on the object's revision code.

**Table 3-2 Revision Code Resulting from Duplicating an Object**

<b>Original</b>	<b>Duplicate with revision code</b>	<b>Duplicate with no revision sequence or code</b>	<b>Duplicate with a revision code but no revision sequence</b>
The original has a revision code	Name of duplicate - same as original file  If original is released: If duplicate is in-work, the revision level of duplicate must be GREATER than revision level of original. If duplicate is released, the revision level of duplicate must be GREATER or LESS THAN revision level of original. If original is in-work: You CANNOT duplicate if the duplicate would be in-work. Name of duplicate - different than original file. You CAN duplicate.	Name of duplicate -- same or different than original file You CANNOT duplicate.	Name of duplicate - same as or different from original file You CAN duplicate.
The original does not have a revision sequence or a revision code.	Name of duplicate - same as or different from original file You CAN duplicate the file if you specify a revision for the duplicate.	Name of duplicate - same as original file You CANNOT duplicate.	Name of duplicate -- same as original file You CAN NOT duplicate Name of duplicate - different than original file. You CAN duplicate.
The original does not have a revision sequence but it does have a revision code.	Name of duplicate - same or different than original file You CAN duplicate the file if the revision level of the original is valid for the revision sequence of the duplicate.	Name of duplicate - same or different than original file You CANNOT duplicate.	Name of duplicate - same as original file You CAN duplicate if original is released. You CANNOT duplicate if the original is in-work. Name of duplicate - different than original file. You CAN duplicate.

## Status Level

A Vault object (file, part, or binder) evolves along a predefined path from the time it is stored until it is reviewed and released. Vault indicates the object's state of completion by associating a sequence of status levels with it.

Each set of status levels composes a revision. The revision code of a Vault object distinguishes the evolutionary stage of that object.

The Status Level indicates an object's state of work and user accessibility. An object passes through a standardized sequence of status levels within each Revision, as defined by the Vault Administrator. As the review and approval process proceeds, Vault updates the object's Status Level as necessary to reflect its current status.

When a stage of work on a file completes, a review cycle commences to determine if the file is ready for the next status in the designated authority scheme. Once an object has a released Status Level, it can not be modified.

If you do not specify a Status Level when you store an object, Vault assigns the initial in-work Status Level.

## Classification

An object's classification determines ownership and the authority scheme. You designate the file Classification when you store a file in Vault. A file can have only one Classification:

- A Private object is owned by a Vault user
- A Project object is owned by a project
- A Public object has no designated owner

Anyone can access a public file. A private file belongs to the user who created it. Only the owner of a private file can read and modify it. Since a project file belongs to a particular project, it is controlled according to the authority scheme for that project.

## Project

A project is a set of files and parts that can be defined and managed as a unit. While each project must have its own authority scheme, that scheme need not be unique. Projects are intended to be used for logical segments of activity at your site.

A project is one of Vault's methods for grouping objects. Once an object is stored and defined as a member of the project, it is owned by that project and operates based on the authority scheme of the project. You can work with files and parts in a project as a unit.

A project has a revision sequence, an authority scheme, and data. The authority scheme controls an assigned user's ability to both access and modify an object

defined as part of the project. While each project must have its own authority scheme, that scheme need not be unique.

A file or part can only belong to a single project.

## Project ID

If you are storing a file or part with the Classification of PROJECT, you must also indicate a Project ID to identify the project to which the object belongs.

## Part Name

If you are adding a file to an existing part, type the name of the part in the Part Name box.

## Local Rule

The Local Rule specifies the client rulebase in which the object is maintained.

- LOCAL indicates that the object is a standalone file.
- CADD5 indicates that the object is a CADD5 part.

Your site may support additional Local Rule options. Check with your Vault Administrator.

## Password

You can assign a password when the file is initially stored. Each password does not have to be unique; many files can share the same password (for example, all files in a project can share a password).

To access a password-protected file, you must enter the correct password. To protect your password's security, the system does not display your password as you type it.

Parts, file sets and projects may consist of files with and without passwords. However, to perform a multiple file transaction on a part or file set, all of the files must have either the same password or no password. Vault does not process a part or file set unless it can access every file in the part or file set.

If you perform a multiple file transaction on a project, Vault processes only those files that have the password you specify (or those with no password if you do not specify one). Vault performs no action on the remaining files.

A password may be up to eight alphanumeric characters.

Please note: Do not enter Kanji characters within a password.

## Storage Pool

A Storage Pool is a UNIX partition, similar to a directory on your local system. It contains the physical files as stored in Vault. The Vault Manager Guide describes managing storage pools.

## Vault Attributes

An attribute is a piece of information related to a file or part. It has a name, type, and value.

Vault Attributes allow you to include information specific to your application. Vault derives no particular information from them; your use of them is both optional and flexible. The following properties are Vault Attributes:

- System type
- Description
- Group Technology Code
- User-Defined Type
- Part Number

You can apply Vault Attributes to any object you can store in Vault.

Please note: The Units of Measure attribute capability is documented in the *Vault Manager Guide*.

Boxes for entering values corresponding to these properties display on the Store In Vault dialog box and the Save As dialog box (which appears when you choose Duplicate from the menu bar).

You can later modify an object's Vault Attributes. The following procedure illustrates changing the Vault Attributes of a file.

1. From the menu bar, choose View > Vault.
2. From the menu bar, choose Edit > File Attributes. The Change Vault Attributes dialog box appears.
3. Edit the Vault Attributes that you want to change.
4. Click OK.

## User-Defined Attributes

You can use User-Defined Attributes to hold additional information about your files. Doing this allow you to maintain an association between important information about your parts and files and the parts and files themselves.

Generally, the Vault Administrator creates User-Defined Attribute properties and associates them with an object. The Vault Administrator then creates an Attribute File which contains a template for entering a value for User-Defined Attributes required by the object.

The Vault Administrator informs you of the name of the Attribute File for each object requiring one. You indicate the value of each User-Defined Attribute in the Attribute File and apply the values in the file to the object with one of three actions:

- Store
- Check in

The *Vault Manager Guide* details creating a User-Defined Attribute.

### Attribute File

Some Vault dialog boxes provide an Attribute File box for the name of a previously created Attribute File. This Attribute File contains values corresponding to User-Defined Attributes. Vault automatically enters these values at one time, if you include the Attribute File name when you store an object or if you specify one when you check it out or in.

Please note: This file is for associating User-Defined Attributes only. You cannot include Vault Attributes in an Attribute File.

If an object requires that you include a value for any User-Defined Attributes, you cannot store the object without specifying an Attribute File name. Conversely, if

an object has no associated User-Defined Attributes (or if the Attribute File you specify contains assignments for User-Defined Attributes not required by the object), Vault does not perform the requested action and displays a message stating that the operation does not satisfy the required attributes.

An Attribute File requires the following lines at the top:

```
OBJECT-NAME=object name  
OBJECT-TYPE=FILE or OBJECT-TYPE=PART
```

where `objectname` indicates the name of the object to which you are applying the values in the Attribute File.

The file should then include one assignment on each line, in the following format.

```
ATTRIBUTE_NAME=attribute value
```

The `ATTRIBUTE_NAME` is always uppercase; `attribute value` can be mixed case. The value immediately follows the equals sign. Vault ignores trailing blanks. An example of this syntax follows:

```
SUPPLIER_NAME=VCM Corp.  
SUPPLIER_PART_NUMBER=74345.4  
TOTAL_WEIGHT=47 KG
```

If an object you store requires attribute values that you have not specified, Vault does not perform the requested action and displays a message stating that the operation does not satisfy the required attributes. This can also occur if the data type or classification of a stored object changes and you try to perform an action on it.

If you check out or check in an object and do not include the name of an attribute file, the attribute values defined when you stored the object (or last checked it out or in) remain in effect.

## Viewing Object Properties

You can view current information before you change an object's properties:

1. From the menu bar, choose Edit.
2. Choose from among the properties which display.
3. The dialog box which appears displays the current settings (with the exception of Change Password).

## Changing Object Properties

If you have the proper authority, you can change the associated properties of the following objects stored in Vault:

- File
- Part
- Project
- File set

From the Distributed Vault view, you can access a registered file or part from any connected Vault (as long as you have adequate privileges) and change its properties.

A number of restrictions apply when you change the properties of an object:

- You must have write access to the object.
- You must be the owner of an object classified as Private.
- You must be on the project of an object classified as Project.
- You cannot change the properties of an individual file that belongs to a part.
- You cannot change the properties of an object that is in use.

You can access and edit some of an object's properties from Edit on the menu bar:

- Attributes
- Classification
- Password
- Revision
- File Status Level
- Storage Pool
- User Attributes (These are called Vault Attributes in this Guide)

Please note: Objects stored in Vault are not necessarily associated with all of these properties.

The following procedure details how to change properties of a file. You can adapt this procedure to change the properties of other objects by indicating the type of object you want from the View > Vault or View > Distributed submenu.

1. From the menu bar, choose View> Vault > Files.
2. From the Enterprise Data Management - Vault Files browser, choose the file that you want to change.
3. Choose a task from the Edit on the menu bar.
4. When the Change dialog box appears, edit the boxes you want to redefine.
5. Fill in the Password, if the file is protected.
6. Click OK.

If you access a Change Attributes dialog box and do not change any attributes, Vault displays a warning message that no attributes were designated for change.

The attributes which you can enter or modify in the fields of the Change Vault Attributes box, consist of:

- System type
- User-defined type
- Description
- Part number
- Group technology code

Please note: Properties modified on the Change Vault Attributes dialog box are different from the User-Defined Attributes accessible from Edit on the menu bar. User-Defined Attributes are described earlier in this chapter and detailed in the *Vault Manager Guide*.

## Changing the Revision

A file, part, or binder keeps its Revision code for its entire design cycle, from in-work through release. When the object is approved for release through the release process, the in-work Status Level changes to a released Status Level, although the object maintains its original Revision code.

If a released object later requires modifications, access it with Vault > Check Out > Lock. Vault does not check out the released object. Instead, it creates a copy of the object, assigning it the next Revision code in the sequence, and an in-work Status Level.

You can only change the Revision code of an object that does not already have one assigned. The revision code sequence (created by the Vault Administrator) associated by default when the object was stored, determines whether or not an object requires a Revision code.

If you have the proper authority, you can change (assign) the Revision code of an object with an in-work Status Level. You can not change the Revision code of a file or part if it is released or it is a member of a part, (unless you change the Revision code for the entire part and all of its files).

## Changing the Status Level

If you have the proper authority, you can change the Status Level of the following objects to a new value which is already defined:

- File
- File Set
- Part

To change the Status Level of a file belonging to a part, you must change it for the entire part.

To change the Status Level of all of the files in a project, or all members of a file set, you must provide the current Status Level. Vault then changes the Status Level for the files corresponding to that value.

Please note: You may change the Status Level to one sequence above or below the current status, even if you do not have write access under the new status.

## Changing the Classification

If you have the proper authority, you can change the Classification of the following objects, provided the object's Status Level exists under the Classification you designate, and the object is not checked out or locked:

- File not grouped into a part
- Part (all files)
- File Set (all objects)
- Project (all objects)

When you change the object's Classification, you do not need any additional authority to access the file under the new classification. However, if you change the Classification from Private to Public, there are some restrictions:

- You must own the private object.
- You cannot change an object to become another user's private object.

## Changing the Password

If you have the proper authority, you can add, change, or remove the Password of the following Vault objects, if they are not signed out or locked:

- File
- Part

Please note: Do not enter Kanji characters within a password.

If the file or part is currently password-protected, you must enter the current password or Vault will not let you change it.

### Warning

You can not operate on a part as a whole unless all files in the part have the same password. Therefore, although it is possible to change the password for an individual file in a part, exercise caution when doing so.

## Moving a File to a Different Storage Pool

Storage pool is the Vault term for the location of user files which are under Vault control. If you have the proper authority, you can move a file from one storage pool to another one that already exists in Vault.

# Changing Properties of Grouped Objects

The following table details the results of changing the properties of grouped objects.

**Table 3-3 Changing Properties of Grouped Objects**

<b>Object</b>	<b>Result of Changing Properties</b>
Part	If any of the individual part files in a specified Part cannot be changed (because they are signed out or locked), none of the part files are changed.
Project	If you select all project files (of a specified project), all of the available files within the project are changed and Vault sends a message regarding each unavailable file to the audit file.
File set	If a file set is selected to be changed and any of the member files cannot be changed (because they are signed out or locked, for example), none of the file set members are changed.
Password-protected files	<p>Only those files with the password you specify and those that are not password-protected are changed. You cannot change a group of files with varying passwords.</p> <p>If your Vault User ID has administrative privileges, you do not need to specify a file password.</p>



# Working with Objects

---

This chapter provides in-depth information about checking files in to Vault.

- Storing an Object in Vault
- Checking an Object out of Vault
- Checking an Object Into Vault
- Duplicating a Stored Object
- Marking an Object for Deletion or Archive

Please note: You can find the basic procedures for these tasks in Chapter 2, “Accessing Vault.”

## Storing an Object in Vault

Before you can work with an object, it must be stored in Vault. Chapter 2, “Accessing Vault,” discussed storing local files and file names in Vault. You can adapt the procedure presented there when you store the following local objects:

- File
- Directory
- Part

The browser displays the objects available in different views. When you store an object, Vault deletes it from your local storage area, and you will no longer see it in that view. When you check the object out, Vault will display the copy in the browser’s local view.

When you switch to the Vault view for the object stored, the browser displays it, ordered according to a structure set by the Vault Administrator. The object appears in this view, whether it is checked out or in, unless you delete it.

You can direct Vault to keep a copy of the object in your local directory by clicking the Leave Signed Out button in the Store in Vault dialog box. The default is `off`.

### Storing a Local File

To make a local file available to other users, you must move it from your local storage area to Vault. When you store a file, you can define its properties, although the Local Rule must be Local.

Once you store a file, you can group it with other stored files to create file sets and binders. Grouped objects are related, and can take advantage of greater functionality defined in Vault.

The Object Name box appears at the top of the Store in Vault dialog box. Information in this box defaults to the item highlighted on the browser.

You can rename the file when you store it by typing the new name in the Object Name box. Otherwise, the Vault file name defaults to the local file name (although capitalization changes may occur, depending on the setting of the EDMCASE environment variable in your EDM.DEFAULTS file).

If an object with a released status is stored in Vault (but not in the archive table) with the same name as a local object you want to store, you can store the local object using one of the following techniques:

- Store your object with an in-work status, with a revision greater than that of the object already stored.
- Store your object with a released status, with a revision either greater than or less than that of the object already stored in Vault.

You cannot store an object with the same revision as one with the same name which is already stored.

If an object with an in-work status is stored in Vault with the same name as a local object you want to store, you can store your local object with a released status and a revision level less than that of the already stored object.

Please note: You cannot store an object if an object with the same name exists in the archive table at the same or higher revision.

## Reserving a File Name in Vault

You can reserve a name and set some properties in Vault for a file that you have not yet stored. No physical data is associated with the reserved file name and no file is actually transferred to or stored in Vault with this process.

When you later store the file which uses that name, it must have the same properties as the reserved file name. If you want the file to have different properties, store the file and then change them. If you discover you no longer need the reserved file name, you can delete it from within Vault.

You can work with the reserved file name as though it were an actual file stored in Vault. When you are later ready to associate a local file with this reserved file name, use the Vault > Store process.

## Storing a Local Directory

By storing a directory you can load large numbers of existing standalone files into Vault. However, this process does not create or retain any relationships among the files checked in once they are in Vault, and the directory itself is not stored.

When you store a directory, Vault stores the files from the directory specified in the Object Name box of the Store In Vault dialog box.

If Vault can not transfer a file in the directory, it excludes it and continues to store the remaining files in the directory.

You must observe the following when storing a directory in Vault:

- Vault ignores the Part Name field.
- Vault assigns the same Revision, Classification, Status Level, Password, and Project Id (when applicable) to each file in the directory.
- Vault applies attributes to each file in a directory after it applies the Object Name and Local Rule. Therefore, it can not apply the information for those properties from an Attribute File.

## Storing a Part

Store a part in the same way you store a file. However, when you store a new part, specify the appropriate environment in the Local Rule box. Ask your Vault Administrator which Local Rule to use.

Each part has a unique Object Name, Revision, Classification, and Status Level. Each file belonging to the part share the same values for these properties.

A part can include zero or more files. You can check a part out of Vault only when it includes at least one file.

Your site may have other requirements for storing a new part. Your Vault Administrator can inform you of these.

## Storing a CADDs Part

When you store a CADDs file or part in Vault, you must specify the Local Rule as CADDs.

When you store a CADDs part in Vault, Vault delimits the directories with a dot instead of a slash and converts file names to upper case, regardless of what you specify in the EDM.DEFAULTS file EDMCASE parameter. The part name is also upper case and prefixes each file in the part.

For example, indicating a Part Name of mypart when you store a CADDs part results in Vault storing the following local part files:

```
/user/smith/parts/mypart/_pd  
/user/smith/parts/mypart/_fd  
/user/smith/parts/mypart/draw1
```

as a part with the name MYPART in Vault. Vault also includes the connected files:

```
MYPART.&PD
MYPART.&FD
MYPART.DRAW1
```

You can give the CADDs file or part you are storing a different directory name in Vault when you store it by typing the new directory name in the Object Name box.

For example, indicating an Object Name of PROTOTYPE.BODY when you store a CADDs part results in Vault changing the directories of the following local files:

```
test.body._pd
test.body._nfig.one
```

to the following in Vault:

```
PROTOTYPE.BODY.&PD
PROTOTYPE.BODY.&NFIG.ONE
```

If you do not specify an Object Name when you store an object, the names of the individual files within Vault default to UNIX names with CADDs character conversions. These are shown in the following table.

After you store a CADDs part in Vault, you will see files in the your local CADDs part directory with the suffix `.attr`. These files contain information stored in the Vault database, although the files themselves are not stored in Vault. For details see “CADDs Attributes for CADDs Files” on page 4-6.

Please note: CADDs parts stored in Vault must include an `&PD` file (this is the `_pd` file in your local CADDs part directory). A CADDs part stored in Vault at your site may require other files. Consult your Vault Administrator.

**Table 4-1 Character Conversions for CADDs File Names Copied to Vault**

UNIX Name	EDM Name	Comments
A-Z	A-Z	No change.
a-z	A-Z	CADDs file names are converted to uppercase by the rulebase.
0-9	0-9	No change.
=	=	No change.
/	.	
_ (underline)	&	
.	@	

**CADDS Attributes for CADDs Files:** When a CADDs file or a part, is stored in Vault, each file is assigned CADDs attributes based upon information in the UNIX directory and in the associated `.attr` file for each CADDs file stored.

If the `.attr` file is missing, the file is assigned a file type of 00. For selection scopes of part or files with the file type of 00 are only stored if that file type is included in the CADDs flexible part definition.

## Storing a Custom Part

Your site may have custom parts produced by a CAD/CAM system other than CADDs 5i or by some other software. In some cases, these parts may consist of only one file. In other cases, part files may not share the full part name.

Your Vault Administrator will tell you what to choose in the Local Rule box when you store a custom part to direct Vault when it collects, names, and converts the files belonging to that part. Do not choose LOCAL or CADDs.

## Adding a Local File to a Stored Part

You can only add a local file (one that is not yet stored in Vault) to a stored part when you store the local file.

1. From the menu bar, choose View > Local Files browser.
2. From the browser, choose the name of the file you want to add to the stored part. The prefix of the file name must match the entire part name, and the file name must be longer than the part name.
3. From the menu bar, choose Vault > Store. The Store In Vault dialog box appears.
4. If you have not previously reserved the file name, you can type a value in the System Type, User-Defined Type, System Type, Part Number and Description boxes, but this is not required.
5. Specify a value in the Revision field. If you do not indicate a revision, Vault adds the file to the highest revision of the part.

The revision of the part to which you add the file (highest or explicit) must exist and must not be checked out, archived, marked for archive or for delete, in review, or locked in any other way.

6. Leave the boxes for the following properties blank, since Vault uses the values already associated with the stored part.
  - Classification
  - Project Id

- Status Level
  - Password
7. Type the name of the part to which you want to add the file in the Part Name box.
  8. For Local Rule, choose CADDs if you are working with a CADDs part or the rulebase name provided by your Vault Administrator if you are working with a custom part.
  9. Click OK.

Please note: You may not activate the Leave Signed Out check box.

## Checking an Object out of Vault

When you check an object out of Vault, Vault attempts to name it according to UNIX conventions. The way Vault file converts an object name depends on the type of object:

- File
- Part
- File set
- Binder

and the Local Rule used to store the object.

When you check an object out and lock it, Vault automatically obtains the object's current revision:

- The current in-work revision is transferred.
- If the current revision is released, Vault creates a new revision at the initial in-work status level and transfers this revision.

A name placeholder for the next revision level of the object is created in Vault at the time of transfer. There is no new object in Vault, however, until you either unlock the copy in Vault, or check in and unlock the modified version.

For information on the UNIX file names given to a Vault object when you check it out see Table 4-2, "File Name Conversions for UNIX Files Stored in Vault Checked Out to UNIX" on page 4-11 and "UNIX Names of Checked Out Objects" on page 4-11.

These examples show the input to Local Directory Name to be a relative directory; therefore, the local directory name is prefixed by the current working directory. Vault inserts `$PWD` only when Local Directory Name is not a fully qualified path name, that is, only when Local Directory Name does not start at root (`/`).

If the Vault object name is not a valid UNIX name, Vault does not check the object out and displays a message instructing you to enter a Local Name.

## Checking Out a CADDs Part

When you check out a CADDs part, Vault converts the object using the part name as a directory name and the files that make up the part as files under that directory.

Please note: When creating a new UNIX directory to which you want to copy a CADDs part, Vault starts at the directory specified by the CVPATH logical. If that does not exist, Vault creates the new directory from your current working directory.

When you check out a CADDs part, the CADDs runtime code performs the character conversions shown in the section “UNIX Names of Checked Out Objects” on page 4-11.

## Checking Out a File Set

When you check out a file set, Vault creates a local directory according to the name you specify in the Local Directory Name box of the Checkout dialog box. Vault then copies the parts and files in the file set to that directory. If you do not specify a directory name in this box, Vault stores the files and parts in your local working directory.

## Checking Out a Binder

Checking out a binder is detailed in Chapter 6, “Using Binders.”

## Checking Out Partial Contents of an Object

If Vault cannot check out all of the files in a selected object, the results depend on the type of object. For parts, file sets, or binders, Vault checks out none of the members.

## Naming a Checked Out Object

The name and location of the object in your local storage area depends on the information in the Local Name and Local Directory Name boxes on the Check Out dialog box.

For a single file, the value of Local Name (which defaults to the name you choose in the Vault browser) becomes the local UNIX file name. When you check a single file out of Vault, you can use the Local Name box to rename the file in your local directory.

You can use the Local Directory Name box to specify the pathname to the local directory into which you want the object checked out. If you enter no information in this field, Vault checks your file into your current working directory.

For example, the part `CAR.DASH.AIR` includes these files:

- `CAR.DASH.AIR.&PD`
- `CAR.DASH.AIR.&FD`
- `CAR.DASH.AIR.&NFIG`
- `CAR.DASH.AIR.DRAW1`

If you request a local directory (in the Local Directory Name box) of `edmdir` and a Local Name of `edmpart`, the checked out files in your local directory are named:

- `$CVPATH/edmdir/edmpart/_pd`
- `$CVPATH/edmdir/edmpart/_fd`
- `$CVPATH/edmdir/edmpart/_nfig`
- `$CVPATH/edmdir/edmpart/draw1`

However, when you check out multiple files grouped as a file set or a CADDs part, Vault ignores any value in Local Name and maintains the file names as they were stored in Vault.

Please note: If Vault renamed a file on check-out, you must use the name supplied in Vault to check it in again.

## UNIX Names of Checked Out Objects

**Table 4-2 File Name Conversions for UNIX Files Stored in Vault Checked Out to UNIX**

<b>Vault Name</b>	<b>Requested Local Name</b>	<b>Requested Local Directory Name</b>	<b>Resulting Local File Name</b>
OUTPUT.TXT	my.file	edmdir	edmdir/my.file
	my.file	-	my.file
	-	edmdir	edmdir/OUTPUT.TXT
	-	-	OUTPUT.TXT
MSL.OUTPUT.TXT	my.file	edmdir	edmdir/my.file
	my.file	none	my.file
	-	edmdir	edmdir/MSL.OUTPUT.TXT
	-	-	MSL.OUTPUT.TXT

**Table 4-3 File Name Conversions for CADDs Parts Checked Out to UNIX**

<b>Vault Name</b>	<b>Requested Local Name</b>	<b>Requested Local Directory Name</b>	<b>Resulting Local File Name</b>
CAR.DASH.AIR contains the files:	edmpart	edmdir	
CAR.DASH.AIR.&PD			edmdir/edmpart/_pd
CAR.DASH.AIR.&FD			edmdir/edmpart/_fd
CAR.DASH.AIR.&NFIG			edmdir/edmpart/_nfig
CAR.DASH.AIR.DRAW1			edmdir/edmpart/draw1

**Table 4-4 File Name Conversions for File Sets Checked out to UNIX**

<b>Vault Name</b>	<b>Requested Local Name</b>	<b>Requested Local Directory Name</b>	<b>Resulting Local File Name</b>
FILESET2 contains the files:  PROJ.PLAN (non-CADDS) PROJ.PLAN.SCHED (non-CADDS) LABOR.HOURS.JULY (non-CADDS) NC.ANALYSIS.&BCD.K	-	edmdir	edmdir/PROJ.PLAN edmdir/PROJ.PLAN. SCHED edmdir/LABOR.HOURS. JULY edmdir/nc/analysis/_bcd/k

**Table 4-5 Character Conversions for CADDS File Names Checked Out to UNIX**

<b>Vault Name</b>	<b>UNIX Name</b>	<b>Comments</b>
a-z	a-z	No change
0-9	0-9	No change
A-Z	a-z	Uppercase to lowercase
=	/	First character only
.	/	
@	.	
&	_ (underline)	

## Checking an Object Into Vault

To make the current version of an object available to other authorized users, you must check it back into Vault, whether you have edited it or not. This is true for any checked out object.

- File
- Part
- File Set
- Binder

Some basic rules apply to checking in an object:

- You can check an object into Vault if the object name does not exist in the archive table at the same or higher revision.

The options available when you check an object into Vault are detailed in Table 2-1, “Vault Menu Commands” on page 2-8.

- You can choose Check In > Lock only for objects checked out to you.
- If you reset the lock on individual files belonging to a part, the part remains checked out to you until you reset the lock on the part.
- You cannot reset the lock on private objects owned by another user.
- You cannot change the Local Rule to CADDs when you check an object in, if it was originally stored with a Local Rule other than CADDs.

## Checking In a Part

You must check out an entire part in order to check in any portion of the part. You cannot check out the individual files belonging to the part and then check in the part as a whole.

When you check a part into Vault and do not lock it, Vault processes the transaction as an all-or-nothing file transfer.

- Vault checks in all new member files and connects them to the part.
- Vault removes any member files already stored with the part if the files were deleted locally.

## Checking In a File Set

When you check a file set into Vault and unlock it, Vault only checks in those files in the file set that are checked out. Furthermore, if Vault cannot process any file in the file set (for example, if a file was deleted locally) it will not replace the file set.

Therefore, for file sets that include parts, avoid checking the file set out and locking it or checking it in and unlocking it, unless you are not adding or deleting any part files while the file set is checked out.

You must store all new standalone files into Vault before you can add them to a file set. However, if you add a file directly to a checked in part that is already a member of file set when you store the file, the file set includes it.

## Checking In a Binder

Checking in a binder is detailed in Chapter 6, “Using Binders.”

## Duplicating a Stored Object

You can duplicate any of the following objects from within Vault by choosing Vault > Duplicate:

- File
- Part
- An individual file belonging to a part (without duplicating the part information)

Each of the duplicated files retains the contents and optional file attributes as the original, although you can change them when you request the duplicate. The duplicate retains the original object's source update date. The duplicate must have a unique name or revision level when stored in Vault.

In order to duplicate an object, you must satisfy the following requirements:

- You must have read access to the object.
- You must have write access to the new object (the target). To provide write access, give the new object a status code that your authority scheme allows you to modify.

Vault writes the name of each new file in an audit file. If Vault cannot duplicate one or more files, it reports the number of files that duplicated successfully.

## Duplicating a Checked Out Object

You can duplicate an object which is currently checked out of Vault, although the version you duplicate may not represent the changes made in the checked-out version.

When you duplicate the Vault version of a checked-out object, a message written to the audit file indicates the user who checked out the object.

## Naming the New Object

The name of the new object depends on the type of object you duplicate. The following table shows the default names.

**Table 4-6 Duplicate Object Name Defaults**

<b>Object</b>	<b>Duplicate Name</b>
File	The duplicate has the name you specify in the Object Name box on the Save As dialog box.
Part	Each duplicated file within the CADDs part takes the name you designate in the Object Name box in the Save As dialog box for the new part's name.
	For example, if you copy the FIRST.ONE part to the NEW.ONE part, the copy of the FIRST.ONE.A file takes the name NEW.ONE.A.

## Examples of Names of Copies

The tables titled “Table 4-2, “File Name Conversions for UNIX Files Stored in Vault Checked Out to UNIX” on page 4-11” and “Table 4-3, “File Name Conversions for CADDs Parts Checked Out to UNIX” on page 4-11” which follow present examples of the default names Vault applies to duplicated files, based on the Local Rule and the new copy name.

**Table 4-7 Duplicating the WINGPARTA Part to the WINGPARTB Part**

<b>Name of Original File</b>	<b>Name of Copy of File</b>
WINGPARTA.1	WINGPARTB.1
WINGPARTA.FILE.T3.1	WINGPARTB.FILE.T3.1
WINGPARTA.PART.&PD	WINGPARTB.PART.&PD

# Marking an Object for Deletion or Archive

If one of the following types of objects becomes obsolete on Vault, you can remove it by marking it for either deletion or archive:

- File
- Part
- Project
- File set

You can delete or archive a Vault object by first marking it. Marking the object instead of removing it yourself increases the security of Vault, since the deletion and archive operations occur in a separate step. The Vault Administrator can later perform those actions, which can take considerable time (depending on the file size and system workload).

From the Distributed Vault view, you can access a registered file or part from any connected Vault (as long as you have adequate privileges) and mark it for deletion or archive.

Please note: Binders have different requirements for and responses to these actions. Chapter 6, “Using Binders,” provides information about this functionality.

## Marking an Object

You must first designate, or Mark, the object on which you wish to act. You can similarly Unmark an object if you no longer wish to perform an action on it. For details see “Unmarking an Object” on page 4-19.

If you mark an object, it remains in Vault with restricted accessibility until you or the Vault Administrator performs the indicated action to remove it:

- You can not access an object marked for deletion, except to unmark it.
- You can read an object marked for archiving and you can also unmark it.

To mark an object, perform the following steps:

1. Indicate the object you want to mark it by highlighting it in the scrolled window.
2. To delete: From Administration on the menu bar, select Mark > For Deletion. The Mark for Deletion dialog box appears.

To archive: From Administration on the menu bar, select Mark > For Archive. The Mark for Archive dialog box appears.

3. If the object is password-protected, enter the password.
4. Click OK.

The following conditions determine if you can mark a file:

- You must have write access to the file(s) you are marking.
- To mark a project file, you must be on the project.
- To mark a private file, you must be the file owner.
- You cannot mark a file that is in-use.

If a file is a member of a part, you can mark only the file for deletion. However, to archive a file that is a member of a part, you must mark the entire part.

You can not delete files in a file set until the file set name is removed and the file is removed from all file sets.

## Marking a Password-protected Object

If you are marking a password-protected object, the files which are successfully marked depend on the type of object:

- If you attempt to mark a project that contains files with varying passwords, Vault marks only those files with the password you specify and those that are not password-protected.
- If you attempt to mark a part or file set that contains files with varying passwords, Vault marks none of the files in the part or file set.

Please note: Your Vault Administrator can mark all files in a part, project, or file set even if the files have varying passwords.

## Marking an Incomplete Object

If Vault cannot mark all of the files in a selected object, the results depend on the type of object:

- If you mark a part and any of the individual part files cannot be marked (because they are signed out or locked), none of the part files are marked.
- If you specify all project files (of a specified project) to be marked, Vault marks all of the available files within the project and generates a message in the audit file for each unavailable file.
- If you specify a file set to be marked and any of the member files cannot be marked (because they are signed out or locked, for example), none of the file set members are marked.

## Unmarking an Object

You can unmark an object if the action you requested has not yet taken place. This makes the object available again.

1. Indicate the object you want to unmark by highlighting it in the browser.
2. From Administration on the menu bar, choose Unmark > Selected Object. The Unmark Marked Files dialog box appears.
3. If the file is password-protected, you must enter the password in the box provided.
4. From Mark Type, choose the type of mark you want to cancel: Archive or Delete.
5. Click OK. A message appears stating that the action was successful.

If a marked file is a member of a part that has been marked for archiving or deletion, you can not unmark only that file. You must unmark the entire part.



This chapter provides information about the following tasks:

- Creating a File Set
- Adding a Member to a File Set
- Eliminating a Member from a File Set
- Removing a File Set or a File Set Name
- Working with a File Set

## Creating a File Set

File sets enable you to perform selected file operations on multiple files simultaneously. Many, but not all, operations you perform on files can be performed on file sets. A file set has no associated properties, and the parts and files in one file set are not required to share any properties.

A file set can consist of one or more of each of the following objects in any combination, as long as it is stored in Vault:

- File
- Part
- File set

A file or a part can belong to more than one file set.

You can check out all the members of a file set individually and then check in the file set as a whole. Or you can check out the file set, check in one or more of the members individually and then check in the file set as a whole to replace the remaining file set members.

## Creating and Naming a File Set

You must create and name a file set before you can store files or parts in it.

1. Choose View > Vault > File Sets. The File Sets browser appears displaying existing file sets.
2. Choose File > New to display the Create File Set dialog box.
3. In the File Set field, type the name you want to give the new file set. You can optionally enter information in the Description field.
4. Click OK. The new file set name appears in the browser.

## Adding a Member to a File Set

After you create and name a file set, you can add members to it. Any file set member can be a member of multiple file sets, although recursive file sets are not allowed.

When you check out the file set, Vault checks out READ-ONLY versions of these objects.

When you add a member to a file set, you can add only a single revision. Other revisions of the part or file are not connected in any way to the file set by that single add member procedure.

You do not have to supply the file password to add a file to a file set. However, you may need to supply file passwords when you perform actions on the file sets.

The following procedure describes adding a member to a file set.

1. From the menu bar, choose View > Vault > File Sets. The Enterprise Data Manager - File Sets browser appears.
2. Choose the file set to which you want to add members.
3. Choose Edit > Members from the menu bar. The File Set Member List dialog box appears. If the file set already includes members, they display in the browser.
4. Click Add. The Add File Set Member dialog box appears.
5. Choose values for the fields Member Type and Member Name (editor button is supplied to help you access objects stored in Vault). Type a Description if you want.
6. Click OK.
7. Repeat this until you have included the members you want.
8. Click Cancel to exit the Add File Set Member dialog box.

Please note: When you check a file set out of Vault, Vault expects the file set you check in later to have the same number of files and identical file names. If you add a member to a checked-out file set, Vault will exclude it at check in.

## Recursive File Sets

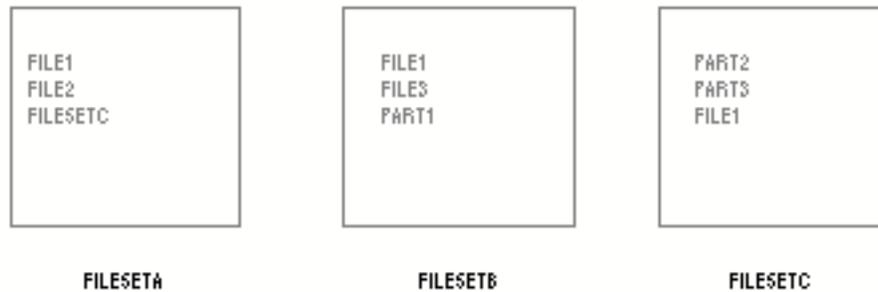
When you designate a file set as a member of another file set, Vault verifies that the parent file set is not a member of the file set you have designated. If the file set designated as a new member includes the parent file set as a member, Vault

displays a message and does not permit its inclusion. This avoids the creation of recursive file sets.

## Including Objects in Multiple File Sets

A file set member can be included in more than one file set. The figure below illustrates three file sets and their respective members.

**Figure 5-1 Illustration of Multiple File Sets**



In these examples,

- FILESETA, FILESETB, and FILESETC are file sets.
- FILE1 and FILE2 are members of FILESETA.
- FILESETC is a member of FILESETA.
- FILE1 and FILE3 are members of FILESETB.
- FILE1 is a member of more than one file set.
- Any modification to FILESETA also modifies FILESETC.
- If one modification fails, none take place.

If you mark FILESETA for archive, FILESETC (as a member of FILESETA) is also marked.

If you mark FILESETC for archive, FILE1 becomes unavailable. Since FILE1 is also a member of both FILESETA and FILESETB, both of these file sets become unavailable, as well.

## Adding a Reserved File Name to a File Set

You can add a reserved file name to a file set. However, when you check out the file set, Vault will not include a representation for the reserved file name, although it will treat the reserved file name as though it were checked out and a message will appear in the audit file stating that the file is a reserved file.

Please note: If a file with the same name as the reserved file name exists in your local storage area, Vault will include that file in the file set when you check it back in.

## Eliminating a Member from a File Set

There are two ways to eliminate members from a file set without eliminating the file set name itself:

- Remove the member objects from the file set one by one
- Delete all of the member objects from Vault by marking the file set for deletion by the Vault Administrator.

## Removing a Member from a File Set

Removing a member from a file set disassociates it from the file set but does not delete it from Vault.

1. From the menu bar, choose View > Vault > File Sets. The Enterprise Data Manager - File Sets browser appears.
2. Choose the file set from which you want to delete members.
3. Choose Edit > Members from the menu bar. The File Set Member List dialog box appears. The current members display in the browser.
4. Select the object you want to remove.
5. Click Remove. The Edmgui Question dialog box appears and asks  
`Do you really want to delete 'filename'?`.
6. Click Yes.
7. Repeat this until you have removed any members you no longer want included.
8. Click Cancel to exit from the File Set Member List dialog box.

Please note: Since Vault checks a file set in with identical member names to the file set checked out, do not remove a member from a checked out file set. If you do, Vault will replace the removed object when you check in the file set.

## Deleting All Members of a File Set

You can mark a file set for later deletion by the Vault Administrator. This procedure deletes the files in the file set, but keeps the file set name.

1. From the menu bar, choose View > Vault > File Sets. The Enterprise Data Manager - File Sets browser appears.
2. Choose the file set you want to delete.
3. Choose Administration > Mark > For Deletion.
4. The Mark for Deletion dialog box appears. Click OK.

5. The Edmgui Information dialog box appears that states:

The fileset has been marked for delete.

6. Click OK.

Please note: You cannot delete a file set while it is a member of another file set.

If you have marked a file set for later deletion by the Vault Administrator, you can cancel that operation before it has taken place by following a similar procedure.

1. When you have chosen the file set you no longer want deleted, choose Administration > Unmark > Selected Object.
2. The Unmark Marked Files dialog box appears. From the Mark Type option list choose Delete.
3. Click OK.
4. The Edmgui Information dialog box appears that states:  
Fileset `filesetname` has been unmarked for delete.
5. Click OK.

## Removing a File Set or a File Set Name

You can remove a file set or a file set name yourself immediately, or mark it for later deletion by the Vault Administrator. You do not have to delete or remove the members of the file set before you remove the file set name. However, if you remove the name of a file set that is not empty, those members still exist.

If you have the proper authorization, you can remove a file set name immediately yourself.

1. From the menu bar, choose View > Vault > File Sets. The Enterprise Data Manager - File Sets browser appears.
2. Choose the file set you want to delete.
3. Choose Edit > Delete from the menu bar.
4. The Edmgui Question dialog box appears that asks:  
`Do you really want to delete 'filesetname'?`
5. Click Yes.

## Working with a File Set

When you work with a file set and its members, you should be aware of Vault's restrictions governing your requests and how Vault implements your direction.

### Checking Out and Checking In a File Set

When you check out a file set from Vault to your local storage area, Vault creates a corresponding copy of each file set member in your local storage area.

- You cannot modify file set members. Vault copies them to your local storage area, however, and you can read them.
- You can check out all members of the file set without locking them in Vault for read access, regardless of the revision levels of the files.
- You can not check out any members of a file set if a single member cannot be checked out (because it is locked, marked for delete, archived, or requires access authority not assigned to you). A success or failure message for each file as well as one for the transaction are written to the audit file.
- You can check individual members of a file set out of and into Vault.
- You can check out all the members of a file set individually and then replace the file set as a whole.
- You can check out the file set, check in one or more of the members individually (specifying either Unlock or Keep Lock), and later check in the file set as a whole (indicating the same lock specification as the individual member checked in earlier) to replace the remaining members.

### Null (or Empty) File Sets

When Vault attempts an operation on a null file set (a file set name with no members) it displays an Edmgui Error box which states:

```
Processing not done - there are no files in the file set or binder  
<yourfilesetname>.
```



Binders help you build a collection of logically related objects. Defining and creating relationships among different objects in Vault allows you to quickly and easily access and review your related collection of data. The following topics are presented in this chapter:

- What Are Binders?
- Object Types and Attributes
- Types of Relationships
- Operations on Binders
- Overview of the Binder Process
- Phase 1: Binder Setup
- Phase 2: Working with Binders
- Adding a Member to a Binder
- Editing Attributes
- Creating Instance Attributes
- Viewing a Binder's Hierarchy
- Removing a Member from a Binder
- Checking a Binder Out
- Checking a Binder In
- Deleting a Binder

# What Are Binders?

A binder is a container for logically related objects. You can use a binder to collect parts, files, or other binders. Once a binder is created, you need to execute only one command to access all objects in the binder. When a change occurs to one of the binder objects, you can easily determine which other objects are affected. Binders can have their own revision and status levels.

Binders can be used for:

- Work packages containing parts and related documents
- Design review folders of drawing objects
- Contract deliverables such as drawings, reports, and documents

**Table 6-1 Example of Binder Content for a Work Package**

<b>Work Package</b>	<b>Binder Content</b>
1. Requirement Analysis	Advance order lists and product specification lists
2. Engineering Design	Advance order lists, product specification lists, and CAD drawings
3. Production Engineering	Advance order lists, product specification lists, and CAD drawings NC output, automatic test data files, stress analysis data, and manufacturing process information
4. Manufacturing	Advance order lists, product specification lists, and CAD drawings NC output, automatic test data files, thermal stress analysis data, and manufacturing process information “As Built” design modification information

Optegra has many base classes of objects such as parts, files, and file sets. Projects, users, and authority groups are also considered base classes of Optegra objects. In the binder application, two new base classes of objects are binder and relation.

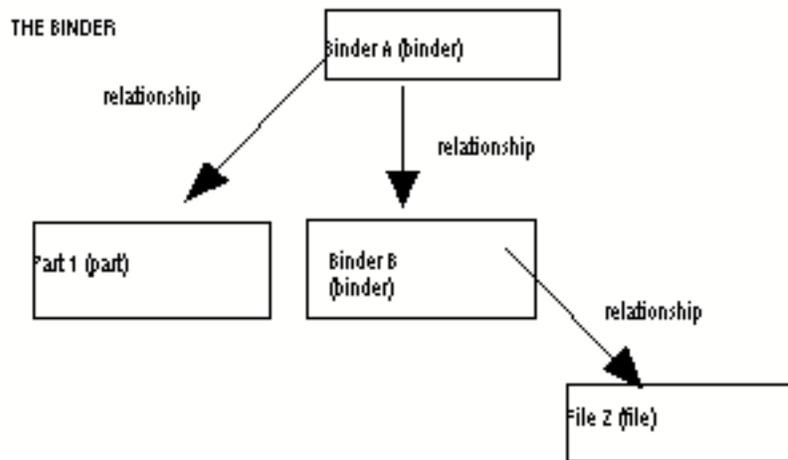
A binder is used to build a collection of objects: parts, files, and other binders. The binder itself is a container for the collection. You create the container and then add objects to it by relating them to the binder. After objects have been connected to the container, the entire collection is referred to as “the binder.”

## What Are Relationships?

Relation (or relationship) is the name of the connector between the container (binder) and the object it contains. Each relationship is defined by the specific parent and child that it relates. The same parent and child can be related more than once. When you relate the same parent and child more than once, you have multiple instances of the same relation.

In the collection shown below, Binder A is called the root binder because it is the top of the hierarchy. It is the parent to Part 1 and Binder B and they, in turn, are the children of Binder A. Binder B is also the parent to File Z.

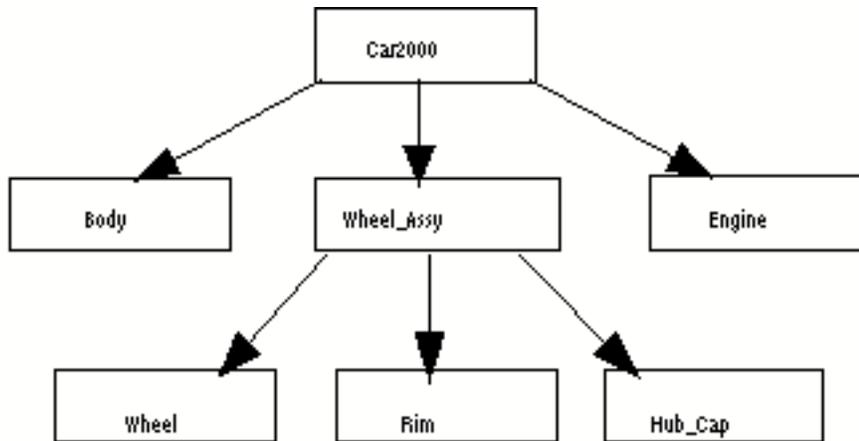
**Figure 6-1 Example of Binder Relations**



# Object Types and Attributes

The following example is referred to throughout this chapter to explain the concepts of binders. The engine is used only in the association relationship example.

**Figure 6-2 Example of a Binder**



A base class is a general category. Before you can create an actual binder or relation, you need to define a specific type of binder or relation subcategory. Specific subcategories give you great flexibility in using binders. Subcategories are called type definitions.

## Example of Object Type Entries

The following table shows the base classes and their object types. These object types are used later in this chapter to create binders.

**Table 6-2 Example of Object Type Entries**

<b>Base Class (Menu choice)</b>	<b>Object Type (User-Defined)</b>
Binder	Wheel_Assy_typedef Body_typedef Tire_typedef Rim_typedef Hubcap_typedef Car_typedef
Relation	Wheel_Assy_relation Body_Assy_relation

## Attributes on Binders and Relations

After you define an object type for the base class binder, you can define attribute groups to use with all binders of that type. Each binder type can have different associated attribute groups (1 to n for each binder) and each group has its own attributes (1 to n for each group). Similarly, you can define an object type for the relation base class and then create attribute groups for the relation.

All attribute groups and their attributes have a unique name associated with them. When you define the attributes, you can choose these data types: integer, real, date, string, or enumeration (a list of values).

### Example of Attribute Group and Attribute Entries

The following entries are defined for attribute group and attribute. They are used later in this chapter to create binders.

**Table 6-3 Example of Attribute Group and Attribute Entries**

<b>Object Type (User-Defined)</b>	<b>Attribute Groups (User-Defined)</b>	<b>Attribute (User-Defined)</b>
Car_typedef	Model_details	Model_name Model_year
Wheel_Assy_typedef	Wheel_Assy_details	Wheel_Assy_number Wheel_Assy_weight Wheel_Assy_date
Body_typedef	Body_details	Body_type Body_material Body_weight
Wheel_Assy_relation	Car_Wheel_Assy	Wheel_Assy_doneby Wheel_Assy_location Wheel_Assy_seqno

Please note: Attribute names must be unique (even across the object type/attribute groups).

Attributes placed on a binder are referred to as class attributes for that binder. Attributes can be placed on any binder. In the following example, the binder Car2000 has a set of class attributes and the binder named Wheel\_Assy has a different set of class attributes.

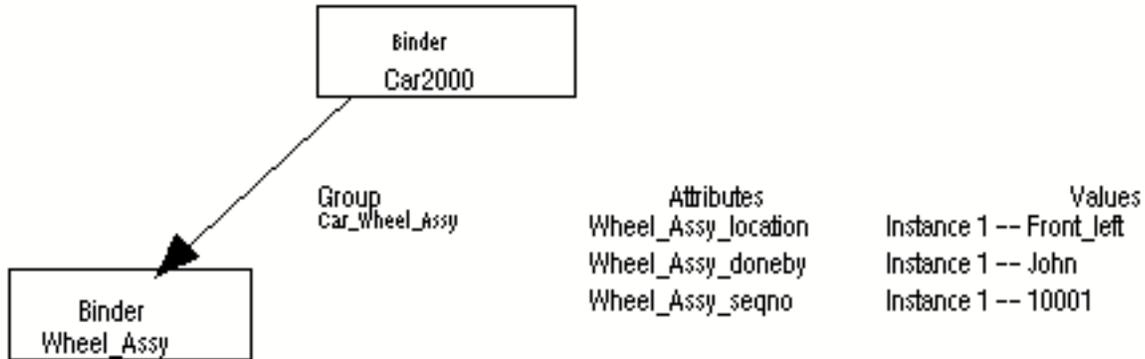
**Figure 6-3 Example of Binder Attributes**



## Attributes on a Relation

You may also place attributes on the relation between a binder and an object it contains. Attributes on relations are called instance attributes. In the following example, Wheel\_Assy\_relation has three attributes (Wheel\_Assy\_doneby, Wheel\_Assy\_location, and Wheel\_Assy\_seqno).

**Figure 6-4 Example of Attributes on a Binder Relation**



Each instance of a particular relation can have different values for its attributes. In the following example, there are four instances of location on the Wheel\_Assy\_relation.

Figure 6-5 Example of Values for Attributes

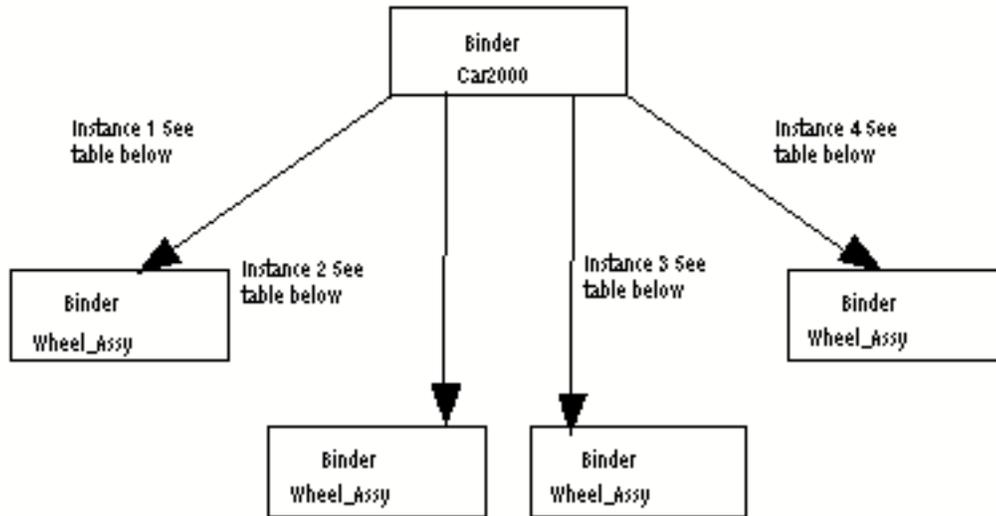


Table 6-4 Values for Attributes in Four Instances

	Object Type	Groups	Attributes	Values
<b>Instance 1</b>	Wheel_Assy_relation	Car_Wheel_Assy	Wheel_Assy_location Wheel_Assy_doneby Wheel_Assy_seqno	Front_left John 10001
<b>Instance 2</b>	Wheel_Assy_relation	Car_Wheel_Assy	Wheel_Assy_location	Front_right
<b>Instance 3</b>	Wheel_Assy_relation	Car_Wheel_Assy	Wheel_Assy_location	Rear_left
<b>Instance 4</b>	Wheel_Assy_relation	Car_Wheel_Assy	Wheel_Assy_location	Rear_right

## Mandatory Attribute Groups and Attributes

When you set up the attribute groups, you can assign whether you want them to be mandatory or optional. The same is true when you set up attributes for the attribute groups. The following table describes what input is necessary when you create one of the four possible combinations.

**Figure 6-6 Description of Input for Attributes**

		Attributes	
		Mandatory	Optional
Attribute Groups	Mandatory	<p>All mandatory attributes in mandatory groups must always be given a value.</p> <p>You must provide a value or a default value must have been specified for the attribute.</p>	<p>Optional attributes in mandatory groups can take the default value.</p> <p>You do not have to provide a value.</p>
	Optional	<p>If any attribute is created in this attribute group (even optional attributes), all mandatory attributes in that group must be given values.</p> <p>You must provide a value or a default value must have been specified for the attribute.</p>	<p>Optional attributes in mandatory groups can take the default value.</p> <p>You do not have to provide a value.</p>

### Example: Mandatory Attribute Groups

The following table is used to explain an example of a mandatory attribute group (Address) whose binder type = Location.

**Table 6-5 Example of a Mandatory Attribute Group**

Attribute Group	Attribute	Optional/Mandatory	Default Value
Address (Mandatory)	Street	Optional	None
	Apartment	Optional	Blank
	City	Mandatory	None
	State	Mandatory	MA

When adding a binder of type Location, values must be provided for the group Address because it is a mandatory group. A successful add object command will always add at least one row of values to the database for attribute group Address.

The contents of a row of values for the group Address are as follows: Street and Apartment are optional attributes. If no value is given for Street, it has a null value in the database. If no value is given for Apartment, it gets a blank value.

City and State are required attributes. If no value is given for City, the command will fail because a value is required and no default is available. If no value is given for State, it will be given the value MA.

## Example: Optional Attribute Groups

The following table is used to explain an example of an optional attribute group (Address) whose binder type = Location.

**Table 6-6 Example of a Optional Attribute Group**

<b>Attribute Group</b>	<b>Attribute</b>	<b>Optional/Mandatory</b>	<b>Default Value</b>
Address (Optional)	Street	Optional	None
	Apartment	Optional	Blank
	City	Mandatory	None
	State	Mandatory	MA

When adding a binder of type Location, values may be provided for the group Address, but it is not mandatory. A successful add object command will only add a row of values in the database for attribute group Address when you provide values for one or more of the attributes in the group.

If you specify any attribute values for the optional group Address, the individual attributes are treated the same way as in the previous example:

The contents of a row of values for the group Address are as follows: Street and Apartment are optional attributes. If no value is given for Street, it has a null value in the database. If no value is given for Apartment, it gets a blank value.

City and State are required attributes. If no value is given for City, the command will fail because a value is required and no default is available. If no value is given for State, it will be given the value MA.

## Multivalued Attributes

Sometimes you need to provide more than one value for an attribute on a binder or relationship. For example, you may wish to keep a record of every time someone tested a binder's design during its development. The tester's name, the date, and a description of the results are recorded. There is only one binder, but many occurrences of the test results.

To maintain such data, you create an attribute group to contain the three items of interest: name, date, and results description. Each time you test the design, you add a new occurrence of the attribute group for the one binder using Edit Class attributes.

If you have other test information that does not need to be repeated with each test, for example, the test plan that is used each time a test is run, do not include that

information in the same attribute group definition, or it will be repeated redundantly with each test result occurrence.

Please note: Multiple occurrences of an attribute group are not the same as multiple instances of a relationship.

# Types of Relationships

You can select two types of relationships for your binder collection. Your choice is based on how you want to do check-outs and reviews. When you create an association relationship, it doesn't matter how the members are life-cycled. When you create an aggregation relationship, there are more rules regarding the life cycle.

- Association —A “loose” collection of the members of a binder.
- Aggregation —A “tight” collection of the members of a binder.

## Association

Association is used when binders are being used to manage a group of objects that are independent of each other or where different pieces of a large project are managed by different groups. It is also used when it doesn't matter if the members have different life cycles. An association relationship allows you to make a change in one member of the collection without affecting the other members.

## Characteristics of Associated Members

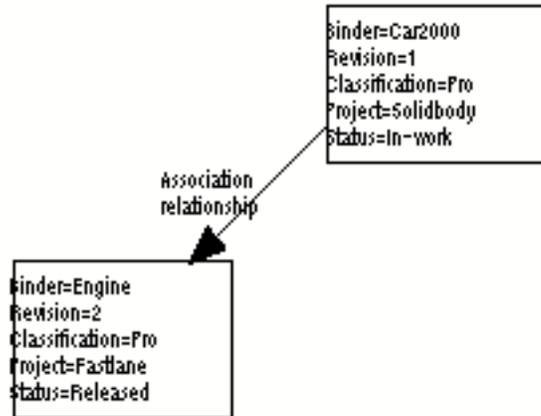
The following are the characteristics of associated members.

- Members can have any project, classification, revision, or status.
- A particular revision must be specified when making the link.
- Members can be aggregated by another binder and they can be members of a part.
- Members can be associated to other parents.
- Members are signed out according to the rules of the binder that aggregates them or the part that contains them.
- Members are signed out when the collection is signed out.
- Members can be obtained for read-only on checked-out of the collection.

## Example

The following example shows part of the hierarchy (the members of the Engine binder are not shown) for a car project with a root binder named Car2000. There is an association relationship between the binder Car2000 and the binder Engine. The engine is developed by an independent project team called Fastlane. The Engine project team is already working on revision 3 of this engine. The Car2000 project is planning to use revision 2 of the engine as part of their project. In this example, the Car2000 team does not control the Engine binder or its members.

**Figure 6-7 Example of an Association Relationship**



## Aggregation

Use aggregation when you are using binders to manage a collection of objects that are dependent on each other. In an aggregation relationship, if you make a change to one member of the collection, the other members are also affected. You use aggregation when you want constraints and dependencies between members.

### Characteristics of Aggregated Members

The following are characteristics of aggregated members:

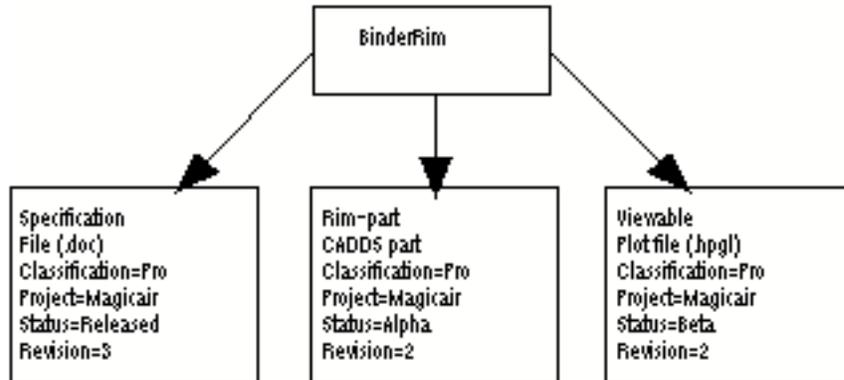
- Members must have the same project and classification (PUB, PRO, PRI) as the parent.
- Members can have any revision.
- The status code may be the same or higher than the parent's status.
- A particular revision needs to be specified when creating the relationship.
- Members cannot be aggregated by other binders and cannot be members of a part.
- Members can be signed out separately from the parent.

### Example 1

In the following example, a binder called Rim is used to collect a CADDS part, the specification that defines what it should do, and the analysis reports performed on it after it completes a phase of work. Because the objects all depend on each other, an aggregation relationship to the binder is used. The specification came

first, so it has the most advanced status code. The part is in a late, but not yet released, status.

**Figure 6-8 Example 1: Aggregated Relationship**



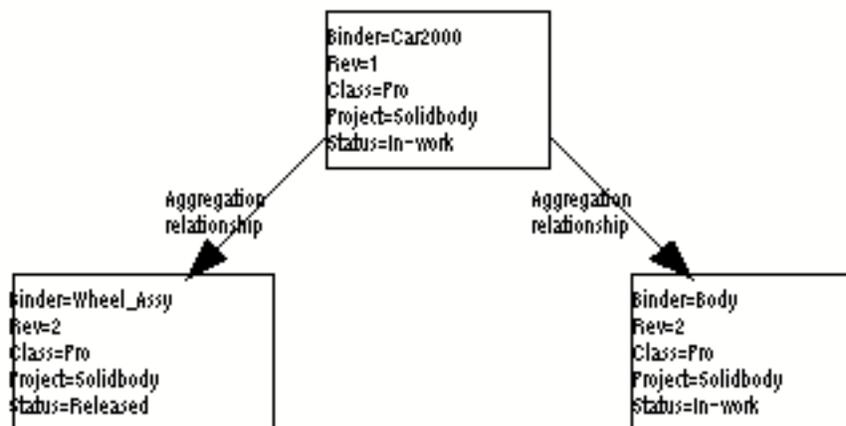
Please note: The aggregated examples may not work if you have a different set of reviewers for the each of the binders. If that is the case, you should consider using an association type relationship.

## Example 2

In the following example, the Body and the Wheel\_Assy are related to Car2000 by aggregation. The body is one that was previously developed by the Solidbody project group and is already on its second revision. It is being developed as part of the Car2000 project and has an aggregated relationship to it.

The wheel assembly is also under control of the Solidbody project, but it is already complete, so has a released status.

**Figure 6-9 Example 2: Aggregation Relationship**



# Operations on Binders

The following operations can be performed on binders.

- Change Classification
- Change Revision
- Change Status Code
- Get
- Read
- Replace
- Reset
- Review
- RSVP
- Sign Out
- Update

## Sign Out

You should be aware of the following when you check out a binder:

- All aggregated members must be available or the check-out will fail.
- If the root binder is in-work, all the in-work part, file, and binder members related by aggregation are checked out.

Associated members are not checked out when the root binder is signed out.

- If the root binder is released, it and all the members that are related by aggregation (which are also released, by definition) are assigned a new revision number and then checked out.

Associated members become associated members of the new binder at their existing revision number.

## Review

You should be aware of the following when you review a binder.

- Only aggregated members with the same status code as the binder are reviewed. Those with higher status codes are not.

Associated members are not reviewed.

# Overview of the Binder Process

The Binder process is a two-phased process—setting up to use binders and working with binders. The table below describes the functions for the steps in Phase 1. Both phases are explained in detail following this overview.

## Phase 1: Binder Setup

The first phase is generally done by your Optegra administrator.

**Table 6-7 Description of Functions for Setting Up Binders**

Function	Description
Define an object type	Choose a base class of binder or relation.
Define attribute group	Enter a name for an attribute group. (One or more for each object type.)
Define attribute	Enter a name and select a data type for an attribute. (One or more for each attribute group.)
Define an enumerated list	If you select an enumeration data type, you must define a name and an enumerated list of values.

The following functions can also be done in the setup phase:

- Delete an object type definition
- Delete an attribute group
- Delete an attribute from a group
- Change an attribute definition
- Add values to an enumerated list
- Remove values from an enumerated list
- Delete an enumerated list

## Phase 2: Working with Binders

The next table describes the functions for the steps in Phase 2.

**Table 6-8 Description of Functions for Working with Binders**

Function	Description
Add binder	Enter a name for the binder.
Add attributes to a binder	Enter a value for the attribute. If the attribute data type is an enumeration, then select an enumeration from a list created in Phase 1.

The following functions can also be done in Phase 2:

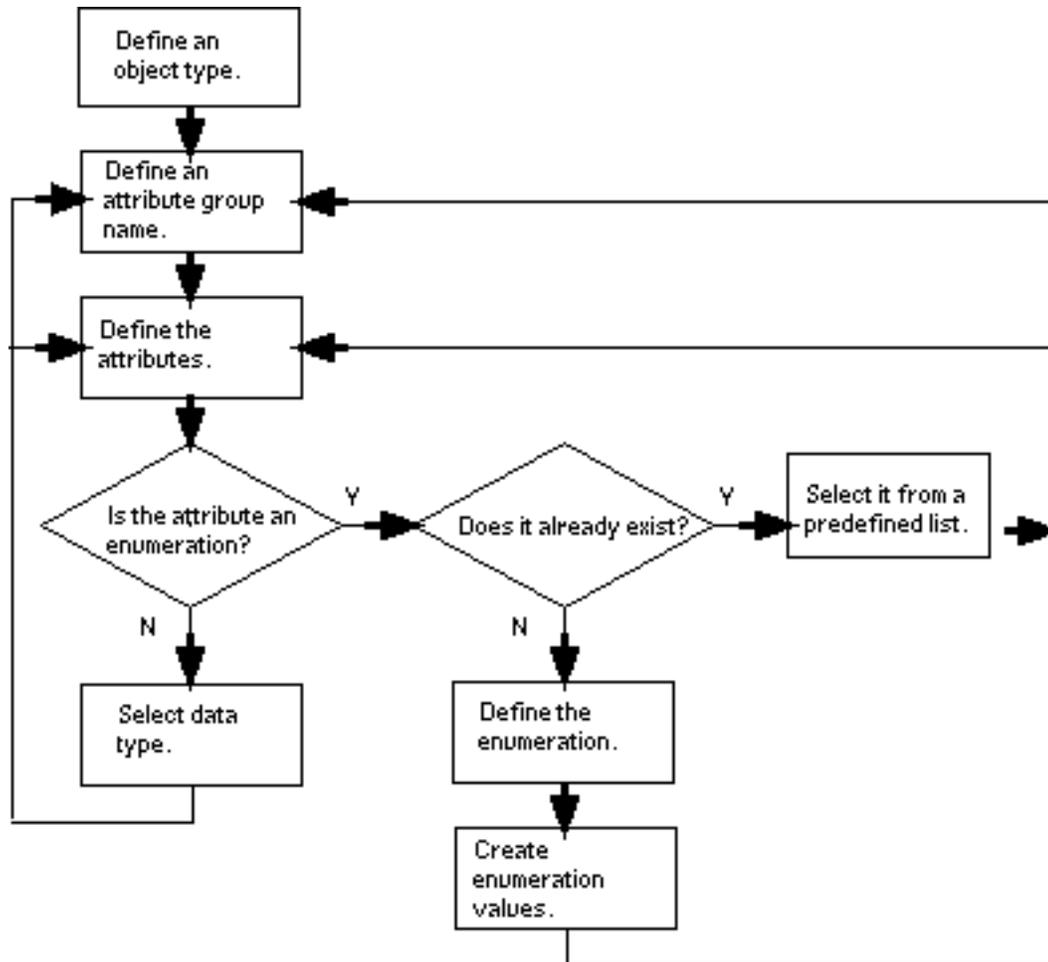
- Change the description of a binder
- Relate binders to other binders/parts/files
- Delete a binder
- Delete a relationship
- List the structure of a binder
- List any dictionary object
- Add attributes to a binder or relationship
- Change attributes on a binder or relationship
- Validate attributes
- Delete attributes from a binder or a relationship

See the next section for details about Phase 1 and Phase 2.

# Phase 1: Binder Setup

The following flowchart shows the process flow for setting up binders. After your pre-planning is done, you can revise where you start in this flowchart by defining the enumeration and creating the enumeration values first. By doing this you can take the path that allows you to select from a predefined list.

**Figure 6-10 Flowchart of the Setup Process**

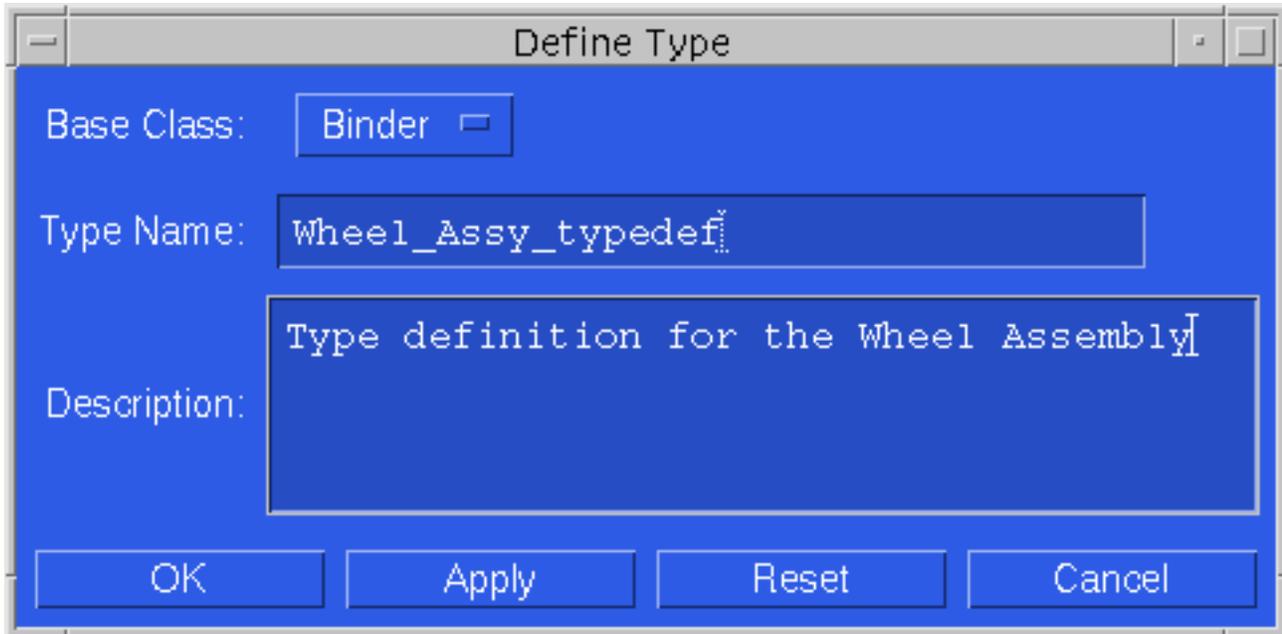


## Defining an Object Type

To define an object type:

1. Select View > Admin > Type Definitions.
2. Choose File > New. The Define Type dialog box appears.

**Figure 6-11 The Define Type Dialog Box**



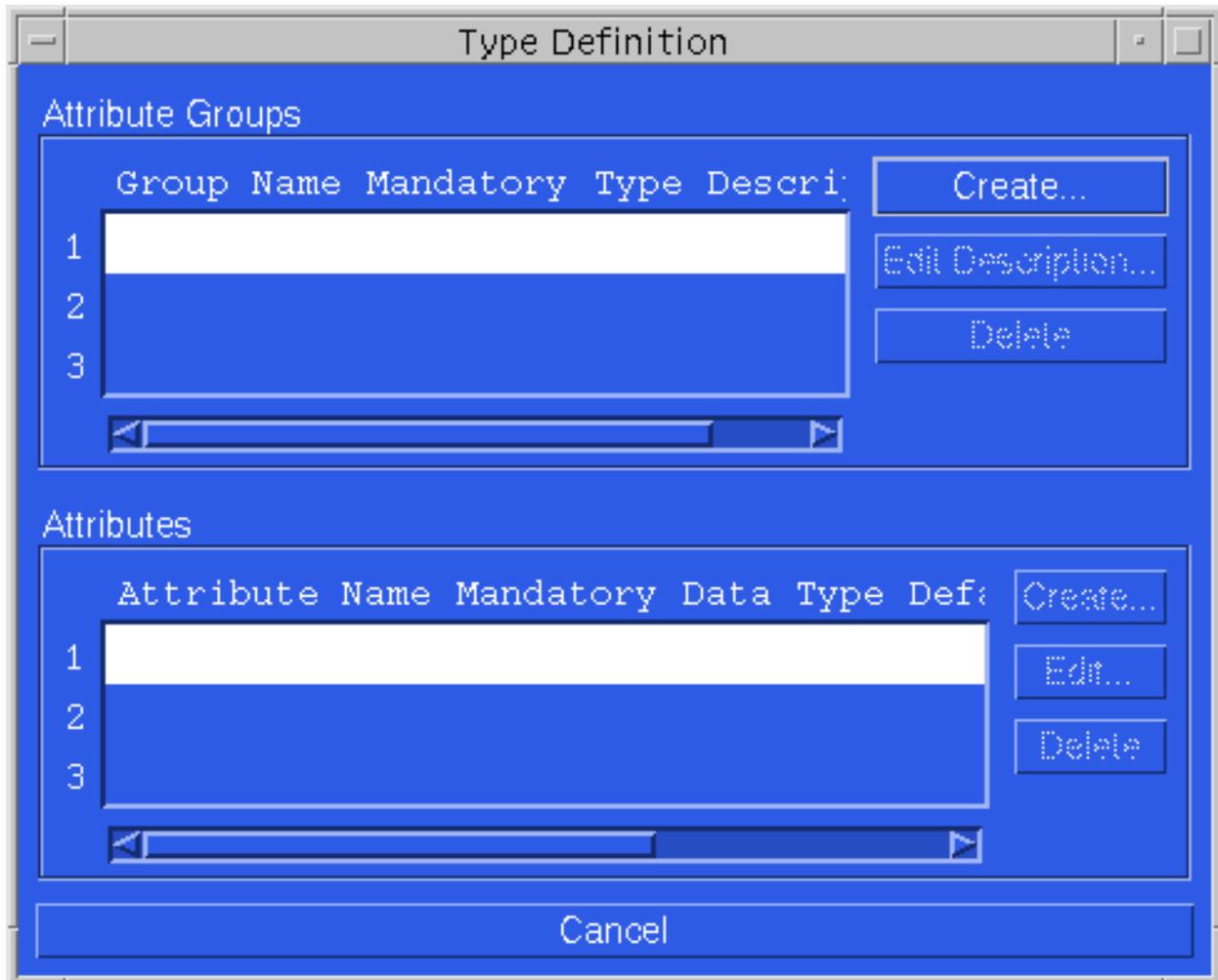
3. From the Base Class option list, choose Binder or Relation.
4. Enter a name in the Type Name text box.
5. (Optional) Enter a description of the binder or relation in the Description text box.
6. Click OK.

## Defining an Attribute Group Name

To define an attribute group name:

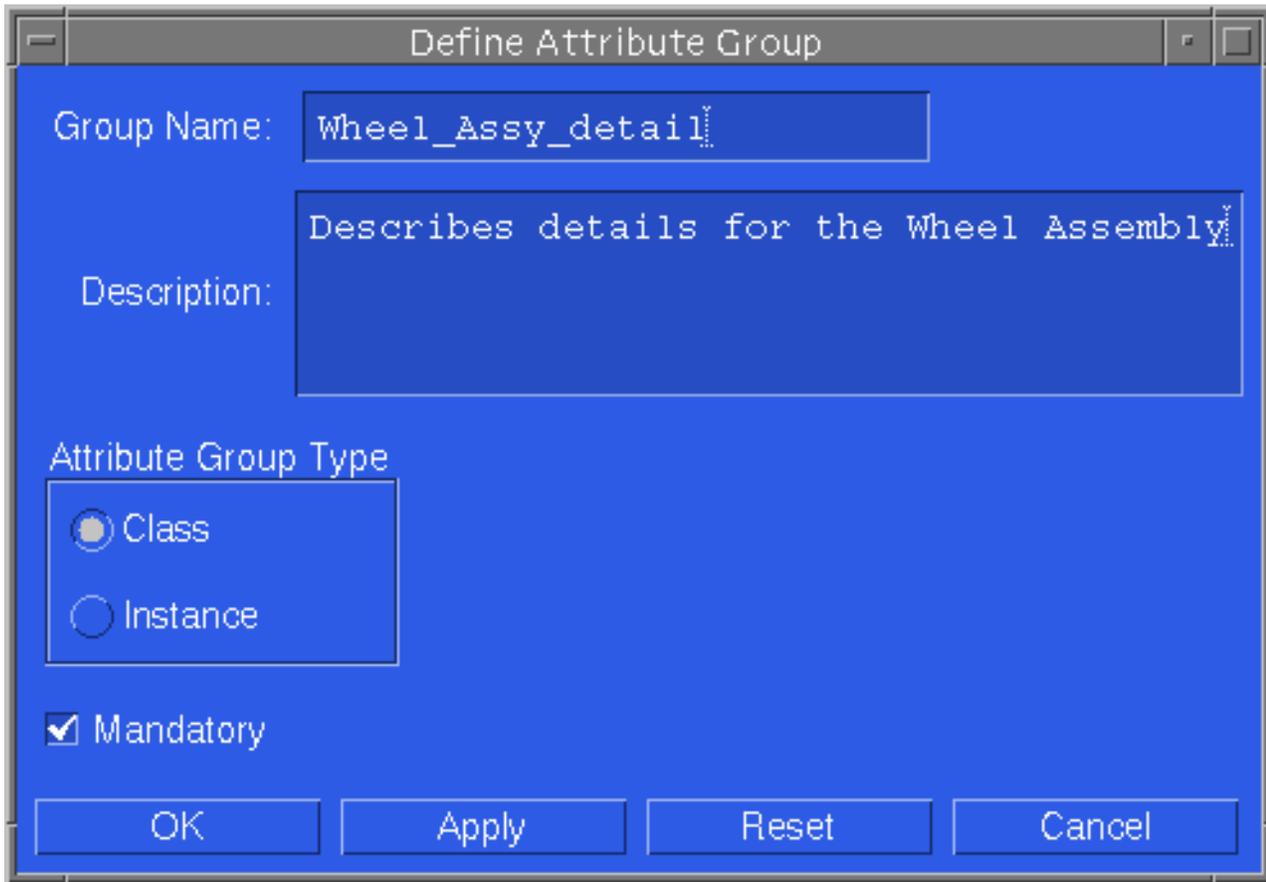
1. Choose the object type for the attributes you want to define.
2. Choose Edit > Type Definition. The Type Definition dialog box appears.

**Figure 6-12 Type Definition Dialog Box**



3. From the Attribute Groups box, choose Create. The Define Attribute Group dialog box appears.

**Figure 6-13 The Define Attribute Group Dialog Box**



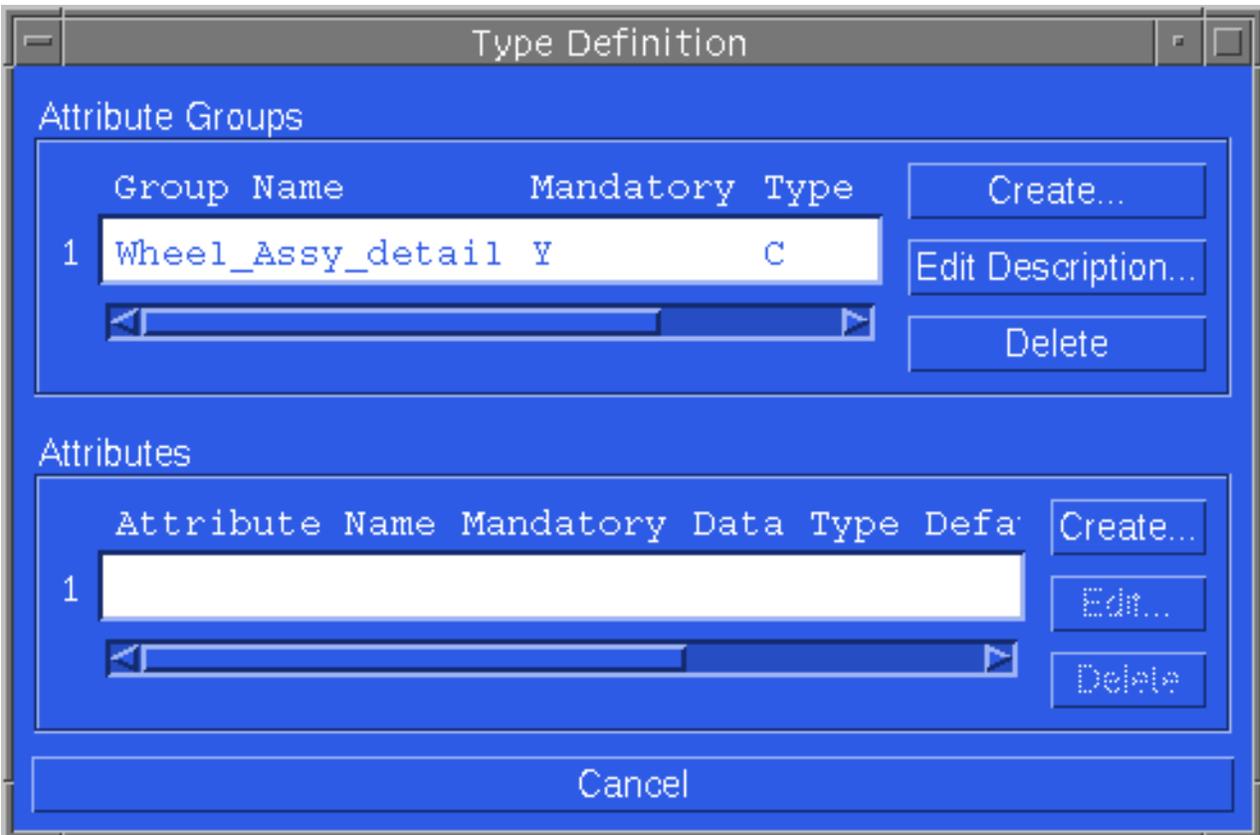
4. Type a name in the Group Name text box.
5. (Optional) Type a description for the group name in the Description text box.
6. Click Class or Instance in the Attribute Group Type box.
  - Class—Creates an attribute group to be placed on a binder.
  - Instance—Creates an attribute group to be placed on a relation.
7. (Optional) If appropriate, click Mandatory.
8. Click OK.

## Defining an Attribute

To define an attribute:

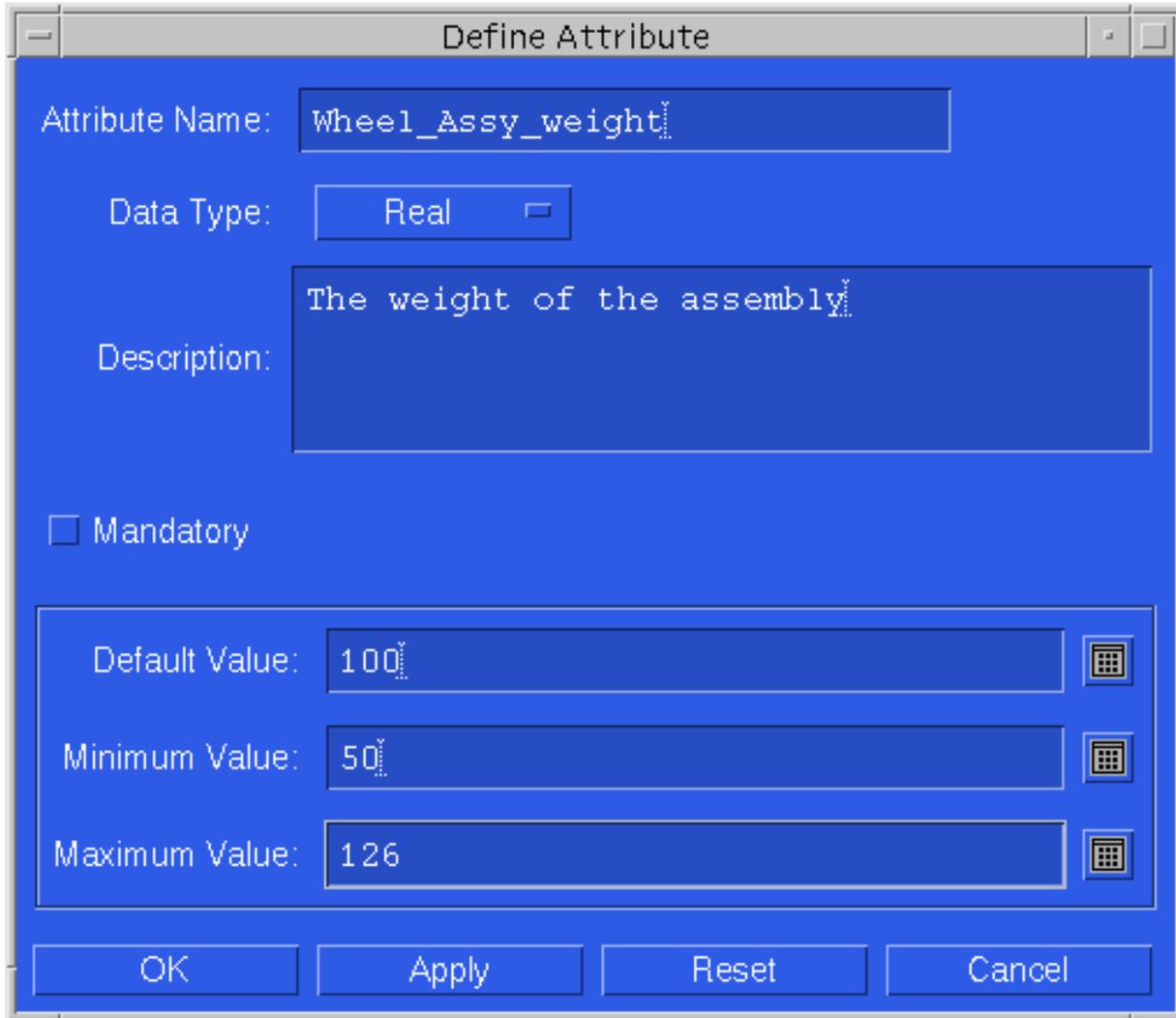
1. Choose the object type for the attribute you want to define.
2. Choose Edit > Type Definition. The Type Definition dialog box appears.

**Figure 6-14 The Type Definition Dialog Box**



3. Select a group name from the Attribute Groups data display area.
4. From the Attributes box, choose Create. The Define Attribute dialog box appears.

**Figure 6-15 The Define Attribute Dialog Box**



5. Type a name in the Attribute Name text box.
6. Choose the data type (string, real, integer, date, or enumeration) from the Data Type option list.
7. (Optional) Type a description for the attribute name in the Description text box.
8. (Optional) If appropriate, click Mandatory.
9. If you select Real enter the values for Default, Minimum, and Maximum fields.
10. Click OK.

## Selecting the Attribute Data Type

You can define five data types in the Define Attribute dialog box.

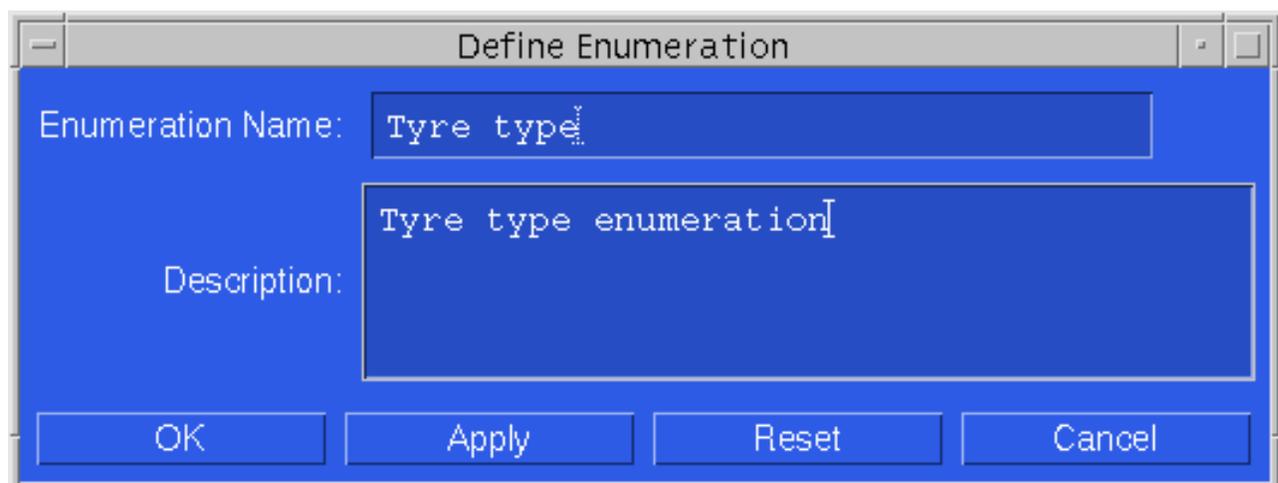
- **String**—The value must be alphanumeric characters. You can also specify a number for the maximum length of characters allowed (255 possible) and a default value for the attribute.
- **Real**—The value must be a real number. You must fill in a minimum and maximum value. You can also specify a default value but it must be between the minimum and maximum values specified.
- **Integer**—The value must be an integer. You must fill in a minimum and maximum value. You can also specify a number for the maximum length of characters allowed and a default value for the attribute.
- **Date**—You must specify a date. You must fill in an earliest and latest date value. They can be entered in any of the following formats: MM/DD/YY, MM/DD/YYYY, DD.MM.YY, or DD.MM.YYYY.
- **Enumeration**—Before using the enumeration data type you should have created the enumeration set. (See [Creating an Enumeration](#).)

## Creating an Enumeration

To set up an enumeration:

1. Choose View > Admin > Enumerations.
2. Choose File > New. The Define Enumeration dialog box appears.

**Figure 6-16 The Define Enumeration Dialog Box**



3. Type a name in the Enumeration Name text box.

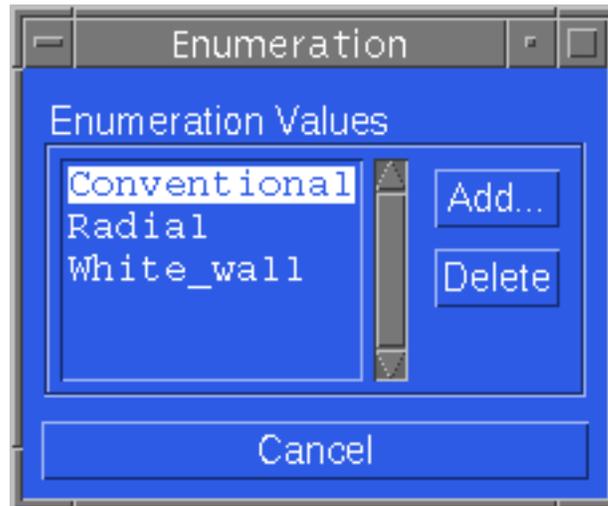
4. (Optional) Type a description for the enumeration name in the Description text box.
5. Click OK.

## Adding Values to an Enumeration List

To add values to the enumeration list:

1. Choose View > Admin > Enumerations.
2. Choose an enumeration value from the data display area.
3. Choose Edit > Enumeration.

**Figure 6-17 Enumeration Dialog Box**



4. Choose Add. The Add Enumeration Value dialog box appears.
5. Enter a value in the Enumeration Value field.
6. Click OK.

## Removing a Value from an Enumerated List

To remove a value:

1. If the Enumeration dialog box is not open, choose Edit > Enumeration.
2. Select an enumeration value.
3. Click Delete.

## Phase 2: Working with Binders

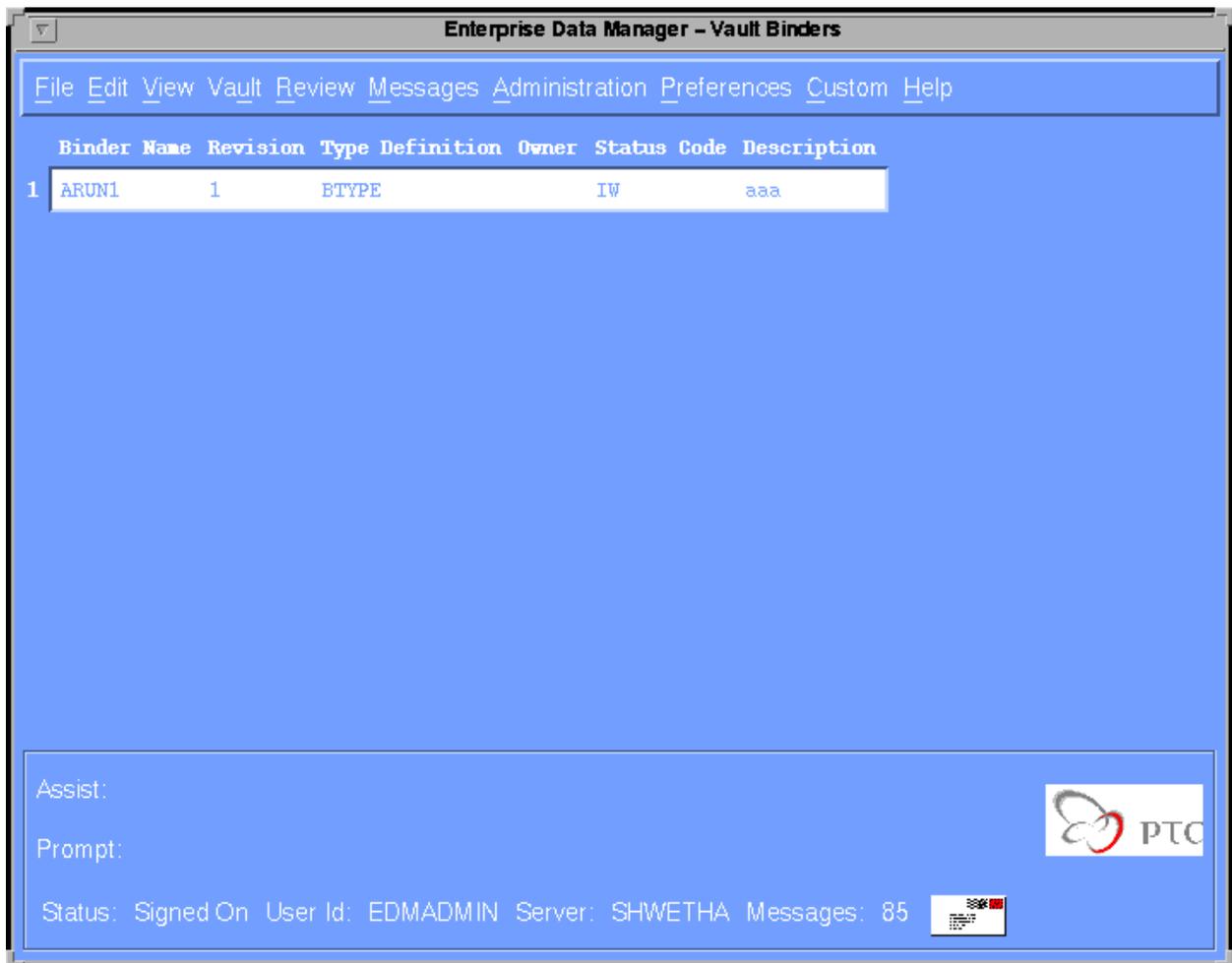
After Phase 1 has been done, you decide whether you want to create a binder or a relation. Then you associate values from the predefined types, attribute group names, and attributes.

### Creating a Binder

To create a binder:

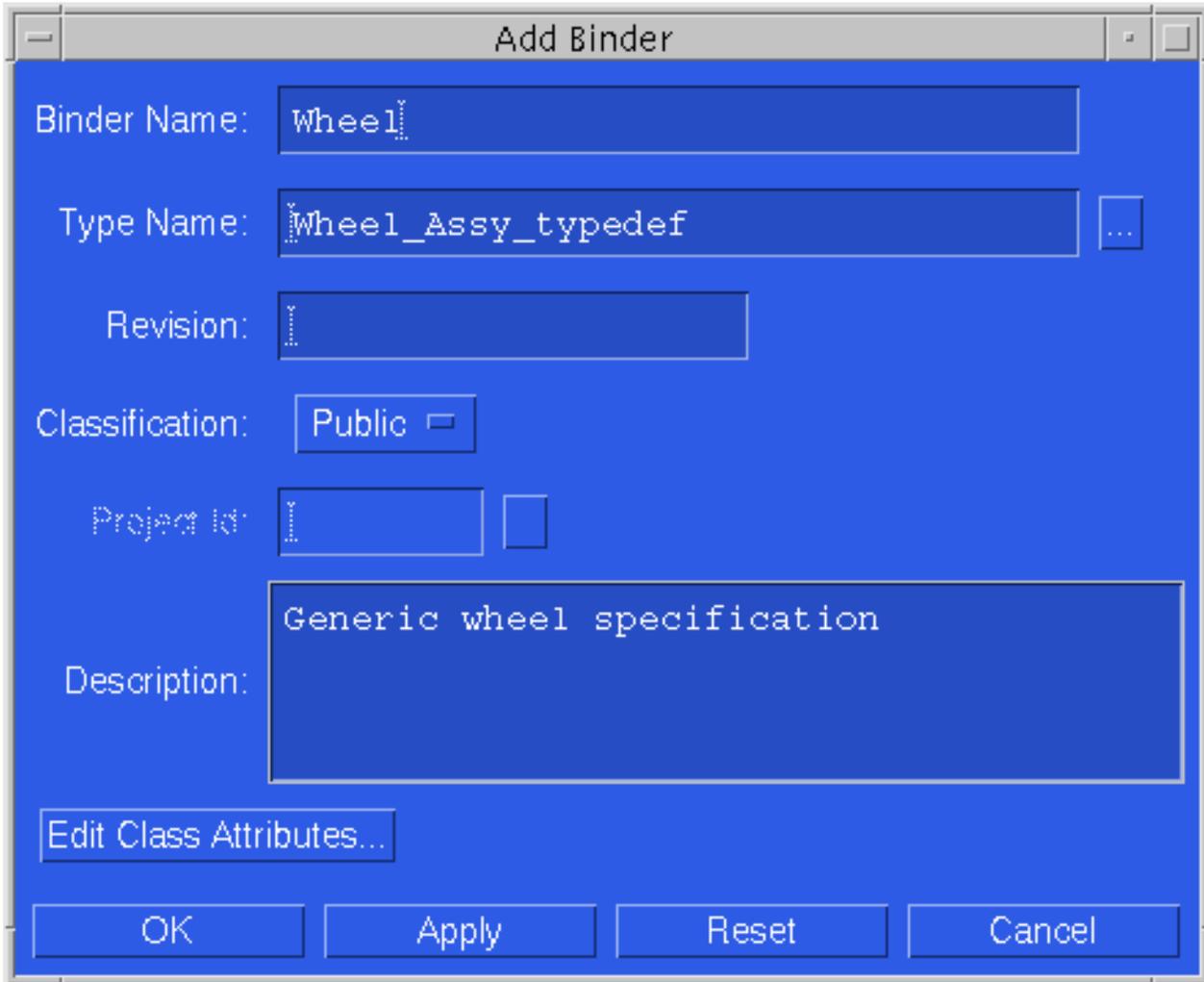
1. Select View > Vault > Binders. If binders have been previously created, a scroll list is displayed.

**Figure 6-18 Vault Binders Dialog Box**



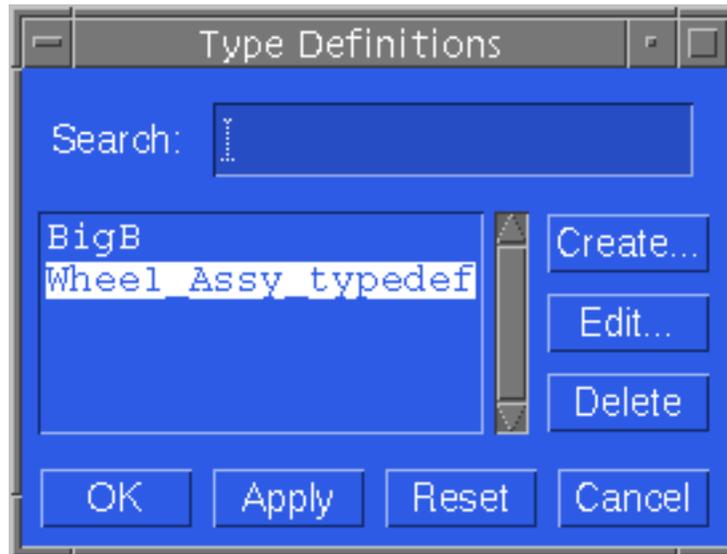
2. Select File > New. The Add Binder dialog box appears.

**Figure 6-19 The Add Binder Dialog Box**



3. Enter a new name in the Binder Name text box.
4. Click on the editor button to the right of the Type Name text box. A scroll list of available type definition names for the base class binder is displayed.

**Figure 6-20 Type Definitions Scroll List**



Please note: If there are no items listed in the Type Definitions dialog box, go to “Defining an Object Type” on page 6-18.

5. Select a type name (type definition) and choose OK to confirm your entry and close the dialog box.
6. (Optional) Enter a description for the object name in the Description text box.
7. To enter class attribute values, click on Edit Class Attributes and follow the procedure in “Creating Class Attributes” on page 6-27.

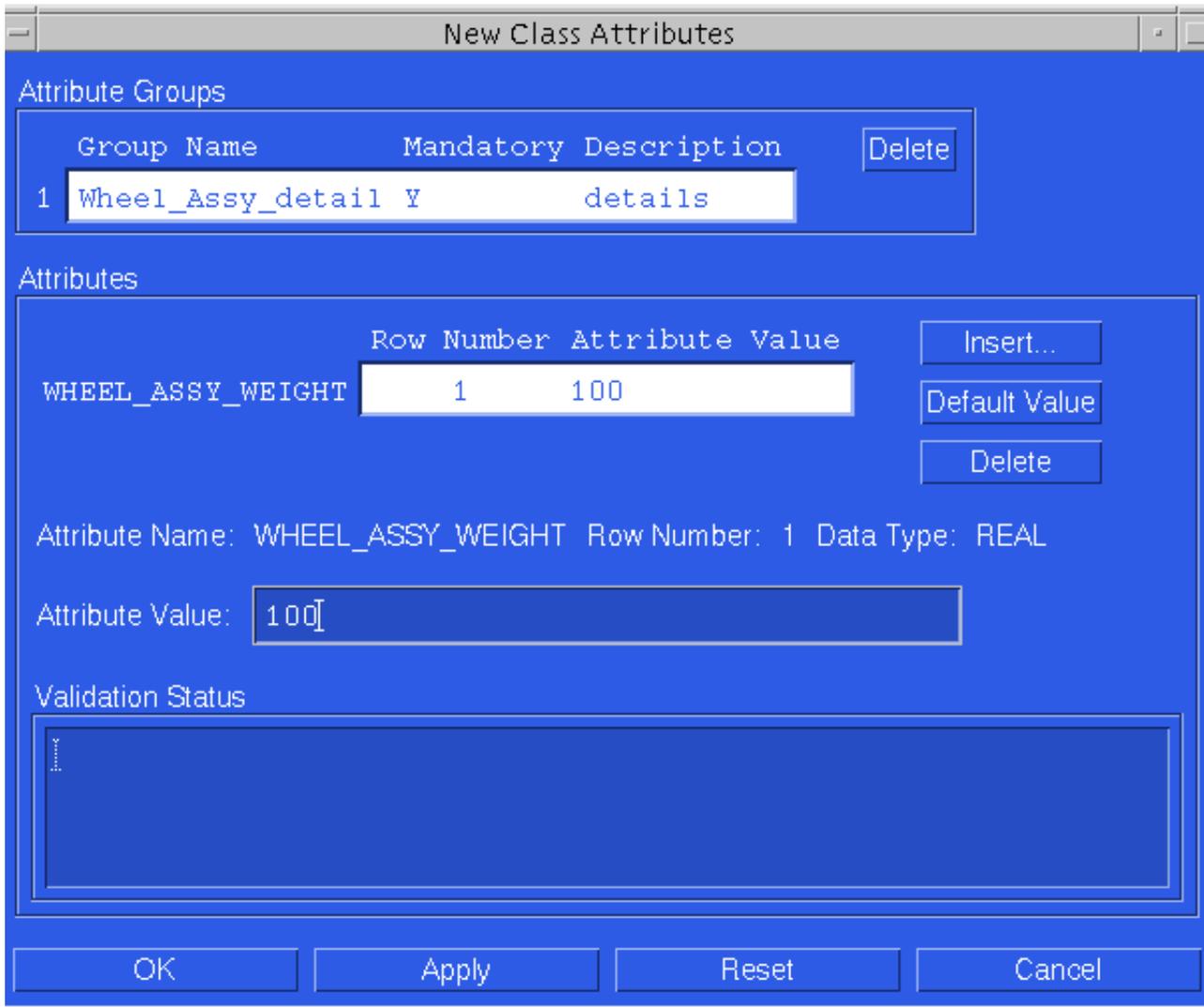
Please note: If the attributes for the attribute group selected is mandatory, then the Edit Class Attributes... bar turns red. See “Mandatory Attribute Groups and Attributes” on page 6-7.

## Creating Class Attributes

To create class attributes:

1. Display the Add Binder dialog box by choosing View > Vault > Binders.
2. Select the desired Binder name, and choose File > New. See “Creating a Binder” on page 6-25.
3. Select the Edit Class Attributes box. The New Class Attributes dialog box appears.

**Figure 6-21 The New Class Attributes Dialog Box**



4. Select the attribute group/attributes whose value you need to enter.
5. Click in the Attribute Value field and enter a value.

Please note: If you click Delete, the value of the selected attribute will delete.

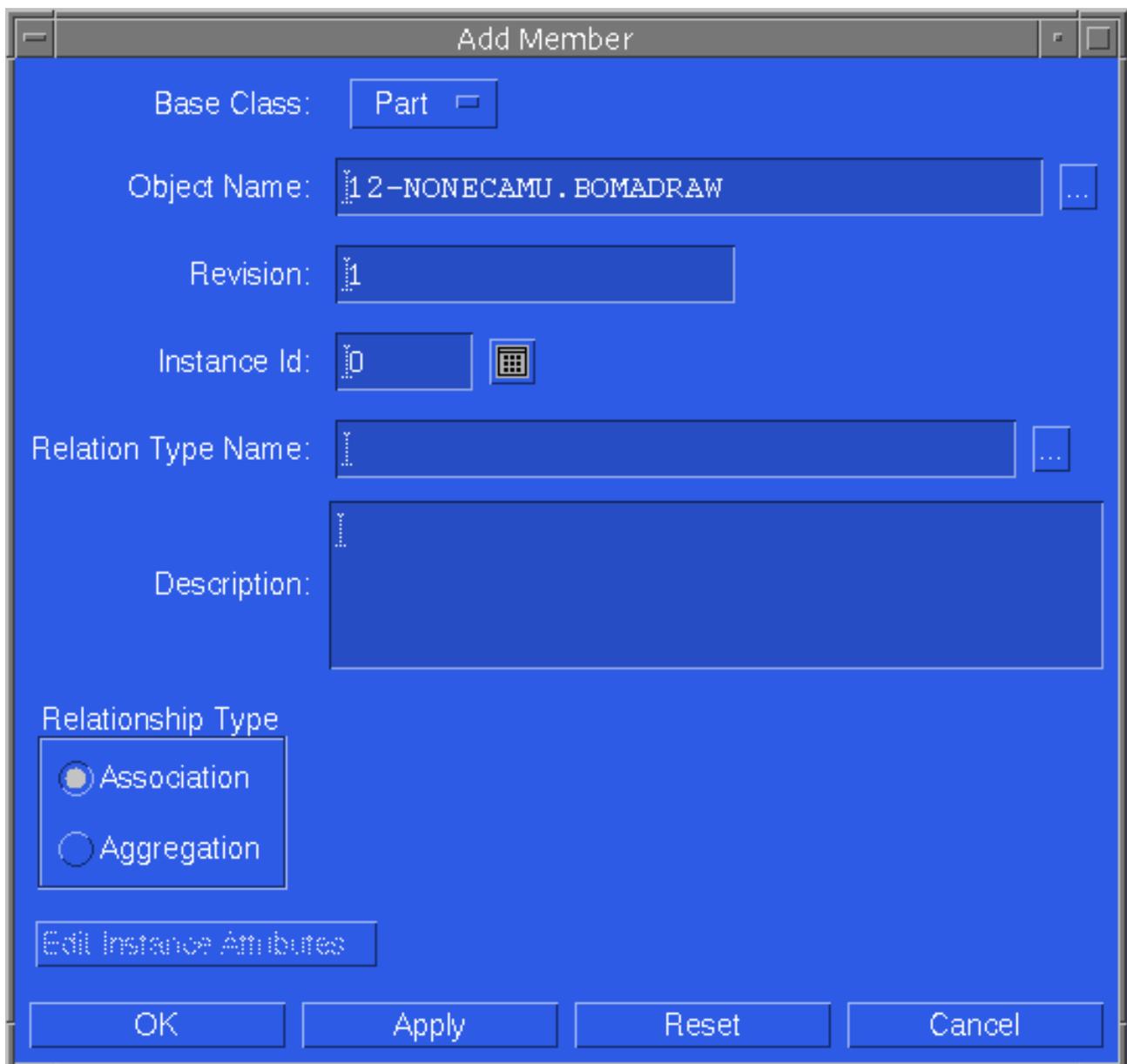
6. Choose OK to process your entry and close the dialog box. Choose Apply to process your entry and have the dialog box remain open.

## Adding a Member to a Binder

To add a member to a binder:

1. Select View > Vault-> Binders.
2. Select the parent binder to which you want to add the members.
3. Select Edit > Binder. The Binder Members dialog box appears.
4. Click Add. The Add Member dialog box appears.

**Figure 6-22 The Add Member Dialog Box**



5. Select File, Part or Binder from Base Class.
6. Enter the name in the Object Name box or click on the editor button to the right of the Object Name text box.
7. Select the desired object from the Vault Objects display.

Please note: If you have a base class of binder, the data displayed is a list of binders. If you have a base class of files, the data displayed is a list of files. If you have a base class of parts, the data displayed is a list of parts.

8. Choose OK to confirm your entry and close the dialog box.
9. Click on the Editor button to the right of the Relation Type Name text box. A scroll list of valid type definitions names for the relation base class is displayed.
10. Select a relation and choose OK to confirm your entry and close the dialog box.

Please note: If there are no items listed in the Type Definitions dialog box, go to “Phase 1: Binder Setup” on page 6-17.

11. (Optional) Enter a description for the object name in the Description text box.
12. Select a relationship type (association or aggregation). See “Types of Relationships” on page 6-11.
13. To enter instance attribute values, click on Edit Instance Attributes and follow the procedure in “Editing Attributes” on page 6-31.

Please note: If the attributes for the attribute group selected is mandatory, then the Edit Instance Attributes bar turns red. See “Mandatory Attribute Groups and Attributes” on page 6-7.

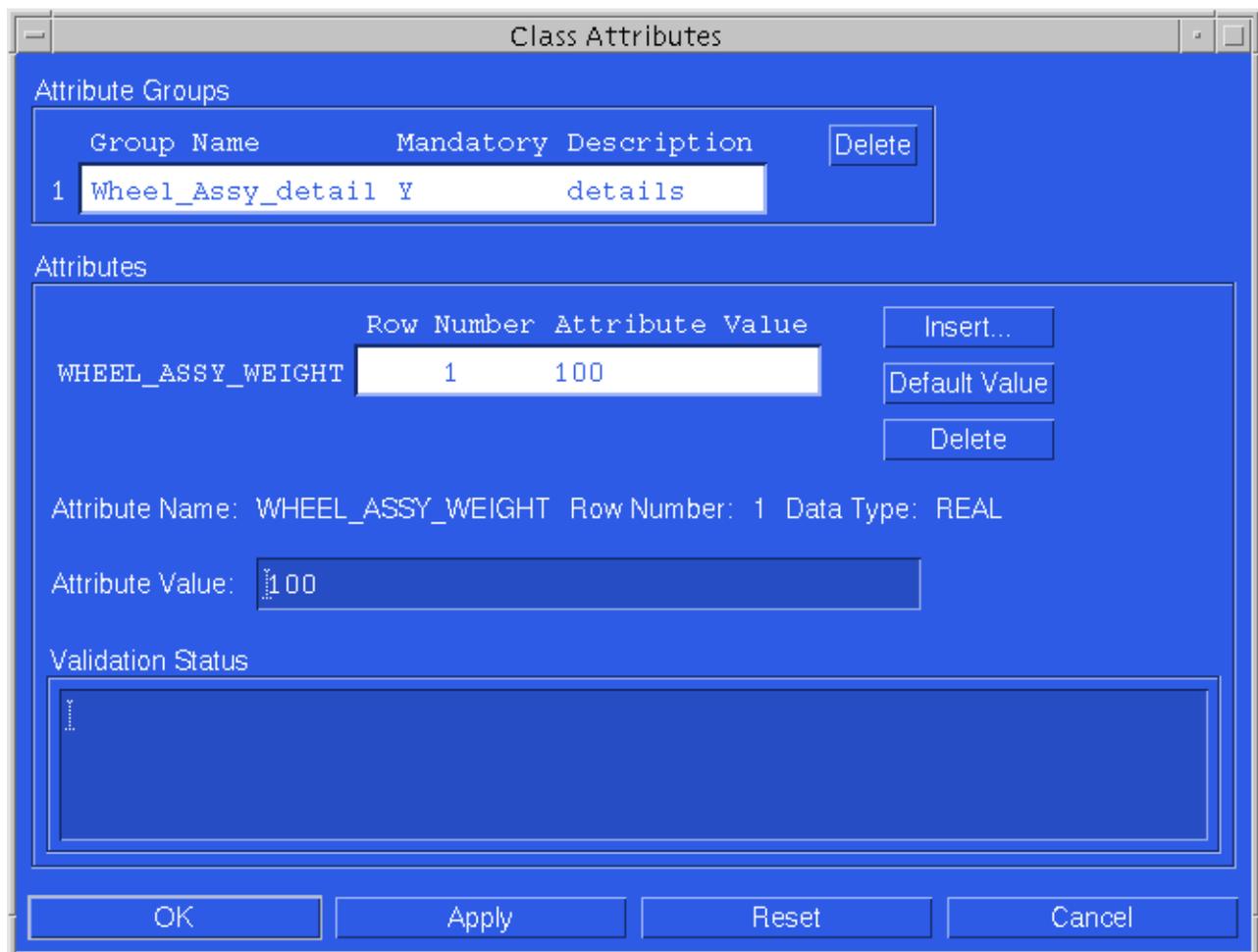
14. Choose OK to confirm your entry and close the dialog box or choose Apply to confirm your entry and keep the dialog box open.

## Editing Attributes

To edit an attribute:

1. Select View > Vault > Binders.
2. Select the binder.
3. Select Edit > Class Attributes.

**Figure 6-23 The Attributes Dialog Box**



4. Select the attribute group.
5. Select the attribute whose value you want to change.
6. Enter a new value in the Attribute Value box.
7. Choose OK to confirm your entry and close the dialog box or choose Apply to confirm your entry.

## Creating Instance Attributes

You can create instance attributes when you add members or edit members.

1. Select the Edit Instance Attributes box in the Add Member dialog box. See “Adding a Member to a Binder” on page 6-29.
1. Select a group name from the Attributes Groups list.
2. Select an attribute from the data display list.
3. Click Insert.
4. Select the attribute name from the data display list.
5. Choose OK to confirm your entry and close the dialog box or choose Apply to confirm your entry and enter more values.

Please note: The attribute is copied in the attribute data display area and it is preceded by the next row number.

6. Click in the Attribute Value box in the Instance Attributes dialog box and enter a new value or select from the enumeration list.
7. Choose OK to confirm your entry and close the dialog box or choose Apply to confirm your entry and enter more values.

## Viewing a Binder's Hierarchy

To view a binder's hierarchy:

1. Select View > Vault > Binders.
2. Select the object you want from the list.
3. Select File > Explode.

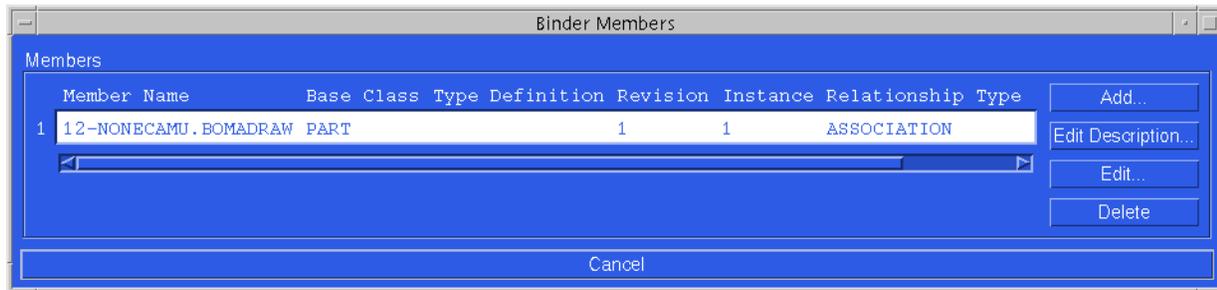
Please note: The contents in the data display area changes based on the base class of the selected object. This means that the data display for an object of the base class FILE is the same as the data display when you choose View > Vault > Files. The data display for an object of the base class PART is the same as the data display when you choose View > Vault > Parts and the data display for an object of the base class BINDER is the same as the data display when you choose View > Vault > Binders.

## Removing a Member from a Binder

To remove a member from a binder:

1. Select View > Vault > Binders.
2. Select the desired parent binder.
3. Select Edit > Binder. The Binder Members dialog box appears.

**Figure 6-24 The Binder Members Dialog Box**



4. Select a member from the list. The member that will be deleted is highlighted.
5. Click Delete.

## Checking a Binder Out

To check a binder out:

1. Select View > Vault > Binders.
2. Select the desired object from the list.
3. To check a file out of Vault for modification, select:

Vault > Check Out > Lock.

To place a copy of an Optegra stored file on your local disk, select:

Vault > Check Out > Do Not Lock.

To check a file out of Optegra without placing a copy on the local disk, select:

Vault > Check Out > Lock Only.

## Checking a Binder In

To check a binder in:

1. Follow steps 1 and 2 of Checking a Binder Out.
2. To check an updated file into Optegra and delete the local copy, select:

Vault > Check In > Unlock.

To update a file in the database without checking it in, select:

Vault > Check In > Keep Lock.

## Deleting a Binder

To remove a binder:

1. Select View > Vault > Binders. A list of binders appears. The first object is highlighted by default.
2. Position the pointer in the Data Display area, then click on the object you want from the list. Doing this highlights the object name.

If the Cascade option is selected, all binders and attributes in that structure are deleted.

Please note: Files and parts are not deleted.

3. Select Edit > Delete.



# Customizing the Interface

---

Vault provides full customization of its graphical user interface to suit your work environment and application needs. This appendix provides guidelines for customization.

- Edmgui Resource File
- Preferences Menu
- Dialog Box Customization
- Font Customization

## Edmgui Resource File

All interface settings are configured within a single resource file named `Edmgui`. You can modify this file to establish the Vault interface settings from two levels:

- User (local resource file)
- System administrator (system resource file)

To customize your interface, use the `Edmgui` resource file and the Preferences menu to make your modifications. You can change the interface in these ways:

- Reposition, add, delete, or rename menu bar items
- Change default values of text boxes
- Hide menu items and/or text boxes
- Change user-visible text

The `Edmgui` resource file lists the default location, order, and name of menu items on the Vault menu. It also tells you which commands and keywords each menu item executes.

You also set CADDs-specific defaults in this file. If you turn off any CADDs defaults, these items do not appear on the interface.

### User Settings

You can set personal interface settings for default values on the Preferences menu or the Design Rule Customization dialog box, shown later. Saved changes are written to the `$HOME/.Edmgui` file. The period (.) in `.Edmgui` distinguishes this file from the system resource file. Settings in `.Edmgui` take precedence over those in the system resource file, described in the next section, but not those in the override directory.

### System Administrator Settings

The system administrator, or whoever manages the Vault at your site, can edit the interface resource file:

```
/usr/apl/edm/data/app-defaults/C/Edmgui
```

## Calling Other Programs from the Interface

With the Vault graphical user interface, you can call executables from the Custom menu for your own programs and software. By adding command lines to the Edmgui resource file, you can customize the interface.

For more information and examples on user-defined entries, check the heading Custom Entry List in /edm/src/data/app-defaults/C/Edmgui. The following sample script launches Vault Navigator from the Custom menu.

```
#!/bin/csh -f
#
# To autoload a local (local disk) ps file.
# NOTE: No command line options to perform autoload exist but it
# can be done thru environ variables. This is the reason for a
# wrapper script to launch navigator.
#
# The entry in the app-defaults/C/Edmgui resource file would look
# like the following:
# Edmgui.StartCN.textValue: Navigator...
# Edmgui.StartCN.source: launch_cn "@SELNAME"&
#
# Unless the user wants to block the Edmgui from continuing until
# the action defined in "source" is complete, the entry should
# always have an appendedampersand so the "system" call will return
# without waiting.
#
# CA_AUTO_LOAD = YES means autoload the ps file defined by
# CA_CONFIGURATION = NO means don't automatic load any files
#
setenv CA_AUTO_LOAD YES
#
# CA_CONFIGURATION is the selname of the file to open. @SELNAME
# argument of entry definition.
#
setenv CA_CONFIGURATION $argv[1]
#
# CA_LOCATION = 1 means file is on the local disk
#              = 2 means file is in the vault
#
setenv CA_LOCATION 1
/usr/apl/edm/scripts/configuration_access
exit(0)
```

## Preferences Menu

The Preferences menu lets you change interface defaults. These local settings and defaults apply only to the appearance and behavior of your own Vault user interface. Examples include colors, text fonts, and printer command settings. From the Preferences menu you can also replace the default query generated for each object type with your own queries.

To change the Vault interface for your personal use, follow these steps.

1. Choose Current Settings from the Preferences menu. (Your entries in the `Edmgui` resource file are displayed.) The Preferences dialog box appears.
2. Edit the text boxes for your desired settings and defaults.
3. Click OK.
4. Choose Preferences > Save Preferences to save your settings to the `.Edmgui` file. (See User Settings under `Edmgui` Resource File for further information.)
5. Exit `Edmgui` and restart in order for the values to be effective.

The Preferences menu items are as follows:

- Current Settings—Opens a dialog box that displays the current interface settings and defaults as currently defined in the `Edmgui` resource file.

Please note: To restore original settings, click the Reset button. OK or Apply implements changes only for the current session.

- Save Preferences —Permanently stores interface changes to your local version of the `Edmgui` file in `$EDM_HOME/.Edmgui`.
- Show Assist Line (and other Show items not shown)—Turns the display for the various items, such as the Assist line, on or off. A selected box indicates that the item is turned on, or active.

Other items from the Preference menu that you can choose to display include Show Status Line, Show Prompt Line, Show Logo Icon, Show Object Status, and Show Tool Palette.

## Dialog Box Customization

The Design Rule Customization box allows you to change the text displayed in the dialog boxes. To change the text in a dialog box, place your cursor on the item you want to change. Press the Shift key and the right mouse button. The Design Rule Customization dialog box appears. Change the text to your preference. If you choose No for Sensitive, the item will be unavailable.

These changes are valid while you are in that session. To have them always valid, choose Preferences > Save Preferences to save your settings to the `.Edmgui` file. (See User Settings under `Edmgui` Resource File for further information.)

# Font Customization

To customize the font, edit the font entries in your `.Edmgui` file.

The syntax to specify a particular font is:

```
<fontname>, <size>, <style>
```

Where:

`<fontname>` is any valid font name available in the X-Server.

`<size>` is any valid size available for a particular font name.

`<style>` can be bold, italic, bold-italic, medium (default). If style name doesn't match with any one of these, it defaults to medium.

In xls format these get mapped to:

```
bold : bold-r  
italic : medium-o  
bold-italic : bold-o  
medium : medium-o
```

These get converted to a standard format specification for font as given below.

```
"*-<fontname>-<style>-normal-***-<size>0-***-***-***"
```

Please note the addition of 0 to the size.

For example, in your `.Edmgui` resource file you specify a font for text field as follows:

```
Edmgui.textFieldFont:courier,14,bold
```

This gets mapped to:

```
"*-courier-bold-r-normal-***-140-***-***-***"
```

Alternatively, you can customize the font from the Graphical User Interface by choosing Preferences > Current Settings from the menu bar and editing the font text boxes which appear at the top.

# Using CADD5 Commands to Access the Vault Database

---

After an introduction to CADD5 5i and Vault, this appendix describes the CADD5 commands listed below. Use these commands to access data in the Vault database.

- Introduction to CADD5 5i and Vault
- GET PART
- PUT PART
- UPDATE PART
- GET FILE
- PUT FILE
- GET FILESET
- PUT FILESET
- Quick Reference for Vault Commands in CADD5 5i

This appendix also includes a Quick Reference for Vault Commands in CADD5 5i.

# Introduction to CADD5 5i and Vault

To use Vault from within CADD5 5i, you must sign on to Vault.

## Starting the DBMS Process

Before you start CADD5 5i and before you sign on to Vault, you must start a DBMS process which allows you to access Vault from within CADD5 5i. The DBMS process must be running under the same user who is running CADD5 5i. To start the DBMS process, enter the following command:

```
dbms &
```

There is no need to terminate the DBMS process after you leave CADD5 5i. Before entering CADD5 5i, verify that the DBMS process is running using the `ps` command at the UNIX prompt. If you reboot your machine for any reason, you must start the DBMS process again before starting CADD5 5i.

## Signing On to Vault from Within CADD5 5i

To sign on to Vault from within CADD5 5i, you enter the `DMPASSWORD` parameter followed by your assigned Vault password when you use the `GET` command (either `GET FILE`, `GET FILESET`, or `GET PART`). Doing this signs you onto the node from which you are getting the file(s). If you do not specify a node with the `NODE` parameter, you are signed on to the default node set up by your system administrator.

You must enter the `DMPASSWORD` parameter with the first Vault command you issue during a CADD5 5i session. You do not need to enter the `DMPASSWORD` parameter again during that session.

Alternatively, instead of entering your Vault password with the first Vault command during a CADD5 5i session, you can include the following line in your `.caddsrc` file:

```
setenv DMPASSWORD password
```

`password` has an eight character maximum. If this line is in your `.caddsrc` file, you do not need to enter your Vault password when you execute the first Vault command in a CADD5 5i session. You must re-enter the password if you change nodes on a subsequent `GET` or `PUT` and the password on the new node differs from that of the previous node.

Please note: To successfully log in to Vault, your Vault User ID and your UNIX login ID must be the same. See the Vault administrator if you have any difficulties.

Please note: Do not use Kanji characters within a password.

## CADD5 Command Description Format

The following format conventions are used to describe each CADD5 command.

### Description

The general purpose, benefit, and use of the command is provided at the start of each command section.

### Valid Modes

This entry specifies the CADD5 level and the mode(s) in which you can use the command. For example, entries may take place within an active part or not in an active part, in Model mode and/or Draw mode.

### Format

This entry shows you the format to use when you type a CADD5 command in its general form. A command consists of the verb/noun combination, modifiers (if any), and `getdata` (if specified). Modifiers provide optional ways of using the basic command. `Getdata` allows you to enter information about locations and to identify entities. Brackets surround optional parts of the command.

### Modifiers

This entry lists and describes the modifiers that you can use within the command syntax. You enter modifiers after the verb/noun combination. Optional modifiers are shown in brackets.

## Examples

Examples clarify the command and its variations using modifiers. Brief written descriptions can accompany examples.

## Vault Commands in UNIX and CADD5 5i

The following four tables show comparisons between Vault commands entered at UNIX level and the corresponding CADD5 commands used to access and manipulate files in the Vault database.

**Table B-1 GET noun (CADD5) and GET (UNIX) Compared**

<b>GET <i>noun</i> (CADD5)</b>	<b>GET (UNIX)</b>
GET	GET
noun	SELSCOPE
PART	PART
FILE SINGLE	FILE
FILESET	FILESET
NODE	DMSERVER (SIGNON)
FILENAME	SELNAME
FILENAME	LFNAME
FILE PASSWORD	FILEPW
Not available	FILETYPE
OVERWRITE	OVERLAY
Not specified (always FILE)	OUTPUT
Not specified (always REPLACE)	APPEND
N/A	CATLEVEL
N/A	DATECRIT
N/A	DATE
N/A	LDIRNAME
NEW	RESERVE

**Table B-2 GET noun READ (CADD5) and READ (UNIX) Compared**

<b>GET <i>noun</i> READ (CADD5)</b>	<b>READ (UNIX)</b>
GET	READ
noun	SELSCOPE
PART	PART
FILE SINGLE	FILE
FILESET	FILESET
READ	Not used
NODE	DMSERVER (SIGNON)
FILENAME	SELNAME
FILENAME	LFNAME
FILE PASSWORD	FILEPW
Not available	FILETYPE
PREFIX	LDIRNAME
REVISION	REVISION
N/A	CATLEVEL
N/A	DATECRIT
N/A	DATE
OVERWRITE	OVERLAY
N/A	CONT
Not specified (always FILE)	OUTPUT
Not specified (always REPLACE)	APPEND

**Table B-3 PUT noun (CADD5) and REPLACE (UNIX) Compared**

<b>PUT <i>noun</i> (CADD5 5i)</b>	<b>REPLACE (UNIX)</b>
PUT	REPLACE
noun	SELSCOPE
PART	PART
FILE SINGLE	FILE
FILESET	FILESET
FILENAME	SELNAME
NODE	DMSERVER (SIGNON)
Not specified (always FILE)	OUTPUT
Not specified (always REPLACE)	APPEND

**Table B-4 PUT noun NEW(CADD5) and STORE (UNIX) Compared**

<b>PUT <i>noun</i> NEW (CADD5)</b>	<b>STORE (UNIX)</b>
PUT	STORE
noun	SELSCOPE
PART	PART
FILE SINGLE	FILE
NEW	STORE
FILENAME	SELNAME
FILENAME (always CADD5 5i format)	LFNAME
NODE	DMSERVER (SIGNON)
Not specified (always CADD5 5i)	ENVIRONMENT
CLASS	CLASS
PROJID	PROJID
FILE PASSWORD	FILEPW
Not Available	FILETYPE
Not specified	STATCD
Not specified	REVISION
Not specified	SYSTYPE
Not specified	USERTYPE
Not specified	FILEDESC
Not specified	PARTNUM
Not specified	GTCODE
Not specified (always FILE)	OUTPUT
Not specified (always REPLACE)	APPEND

# GET PART

Use the `GET PART` command to access a part in the database. No other user can activate, modify, and/or file a part while it is signed out to you for modification.

When you execute the `GET PART` command, the following occurs:

1. The part and its associated files are copied to the local CADD5 5i database volume.
2. The part file is flagged in Vault, preventing other users from modifying the file.
3. The database notes that the part is accessed by your user name with modify access.

The named part and any drawing files, figures, subassemblies, animation or other `TYPE 0` (undefined) files with the same name as the part are copied to your local system when the `GET PART` command is executed. If no part exists under the specified name, and if you choose the `NEW` option, Vault reserves a place in the database for the new part. You receive access to the part and an appropriate message is issued informing you that you reserved a new part. (You can choose the `NEW` option only when accessing the Vault database.)

Please note: `GET PART`, not `GET FILE`, is recommended for part transfers. A `GET PART` request marks all associated files in the Vault database as a group, a `GET FILE` request does not.

## Valid Modes

In CADD5 5i, in or out of a Part or Drawing, in Model or Draw mode.

## Format

```
GET PART partname [modifiers]
```

## Modifiers

```
[PARTNAME] partname
```

Names the requested part. You can enter the modifier `PARTNAME`, but it is not required. You can also enter only the actual part name.

```
[DMPASSWORD password]
```

Signs you on to the Vault node you are getting the file(s) from. You must enter the `DMPASSWORD` parameter with the first Vault command you issue during a

CADD5 5i session. You do not need to enter the `DMPASSWORD` parameter again during that session. Alternatively, you can include the following line in your `.caddsrc` file:

```
setenv DMPASSWORD password
```

`password` has an eight character maximum. You must re-enter the password if you change nodes on a subsequent `GET` or `PUT` and the password on the new node differs from that of the previous node.

[`NODE nodename`]

Names the requested node; the node that you specify is where the `GET` command goes to get the data. Supply both the modifier (`NODE`) and the actual `nodename`. Not required if you are using the default node. Node names can be as many as 18 alphanumeric characters.

```
FILEPASSWORD password
```

Is the password associated with one or more files in the part. Must be specified if passwords exist.

[`READ local directory name`]

Indicates you want read-only access to the named part. The system attaches the local directory name, followed by a period (`.`), to the specified part name, creating a new part name. A copy of the current part is copied to the local disk under the new name. The part is *not* signed out of Vault. If you do not provide a prefix indicating the local directory name, the location of the copy is system specific. VMS and UNIX use your `CVpath` or your current working directory (in that order) to create a new part name. On VM, the part is put on your default minidisk.

Please note: You may optionally include the `PREFIX` modifier immediately following the `READ` modifier and immediately before the value `local directory name`. `PREFIX` is a reserved word that indicates a local directory name follows it.

[`REVISION x`]

Specifies the revision level of the part and is available only with `READ`. A real number, denoting the revision, is required. The default is the highest revision, if released, of the part. CADD5 5i `GET/PUT` commands support only numeric revision codes.

[`NEW`]

Reserves a part name in the Vault database for a part that does not yet exist. This modifier is optional and can be included before or after the part name. The

dependent modifiers PUBLIC, PRIVATE, and PROJECT require the use of the NEW modifier.

[PUBLIC]

A classification group. This modifier allows *any user* with the proper authority to access the part. This is the default.

[PRIVATE]

A classification group. This modifier indicates that you want *only your user name* and the system administrator to have access to the part. A classification group.

[PROJECT identifier]

This modifier indicates that you want *only users assigned to that project identifier* with the proper authority to have access to the part. Both the modifier name PROJECT and the project identifier name must be entered if this modifier is used. The project identifier must exist in Vault.

## Example GET PART WING

```
CDMSON016I Sign on to EDM server NODENAME completed  
successfully. You have 0 EDM message(s).  
CDMGET606I All of the file(s) requested have been processed.  
CDMSOF017I Sign off from EDM completed successfully.
```

In the first example, you gain access to the part. The system response confirms that the part was copied from Vault to your local disk. Notice that when you do not specify the NODE modifier, the default node is assumed.

## Example GET PART WING READ A

```
CDMSON016I Sign on to EDM server NODENAME completed  
successfully. You have 0 EDM message(s).  
CDMREA606I All of the file(s) requested have been processed.  
CDMSOF017I Sign off from EDM completed successfully.
```

In this example you request read-only access to the part. Notice that the system adds your prefix to the requested part when you gained access to it and places the files in a subdirectory called A. This protection prevents you from overwriting the primary version of the original part.

Additionally, you have not locked out other users from this part. Anyone with the required authority can receive a read-only copy of the part, even if the part has not been signed out for modify access.

# PUT PART

The PUT PART command enables you to transfer a part back to Vault. Each PUT PART command, except PUT PART NEW, requires that a GET PART command precede it.

With GET PART, you gain access to the part you want to modify, modify the part, and then return it to Vault. When you return the part with PUT PART, three things happen:

1. The newly modified version is copied back to Vault.
2. The local copy is deleted from the local CADD5 5i workpack.
3. The reference to the access of the part is deleted.

Parts created by the commands COPY PART and CONSTRUCT PART can also be put in Vault even though they have not been signed out with GET PART. If a part of the same name already exists, an overwritten version is *not* permitted by using the PUT PART command.

If you have access to a part in Vault, one of the following scenarios occur:

1. You enter the PUT PART command when you have modify access for the part. Your version of the part is copied back to Vault, overwriting the original part. The local copy is deleted, and your access reference is deleted.

If the part already existed, any drawing files that you deleted on your local version will be flagged for deletion by the system manager and deleted at a later date.

2. You enter the GET PART READ command, naming a part for which you have read-only access. A read-only copy is passed to the local disk with a new name, and the part is not signed out. This copy does not need to be returned to the Control Data Store. If you want to return this part, use the PUT PART NEW command. The NEW modifier is required because the part was given a new name when it was copied to the local disk with the GET PART READ command.

Please note: You cannot execute PUT PART for an active part. Execute EXIT PART F or Q first.

## Valid Modes

Within CADD5 5i, in or out of a part or drawing, in model or draw mode.

## Format

```
PUT PART partname [modifiers]
```

## Modifiers

```
[PARTNAME] partname
```

Names the part to be added or returned to the database. If your part name is the same as a modifier name, enter the word `PARTNAME` before the part name. This no-name modifier allows you to name a part with the same name as a modifier. This naming choice is not encouraged.

```
[DMPASSWORD password]
```

Signs you on to the Vault node you are getting the file(s) from. You need only enter this `DMPASSWORD` once during a CADD5 5i session. Alternatively, you can include the following line in your `caddsrc` file:

```
setenv DMPASSWORD password
```

`password` has an eight character maximum. You must reenter the password if you change nodes on a subsequent `GET` or `PUT` and the password on the new node differs from that of the previous node.

```
[NODE nodename]
```

Names the requested node; the node that you specify is where the `PUT` command goes to put the data. Supply both the modifier (`NODE`) and the actual node name. Node names can be as many as 18 alphanumeric characters. Not required if you are using the default node.

```
FILEPASSWORD password
```

Is the password associated with one or more files in the part. Must be specified if passwords exist. A password is *not* required when adding a new part to the Vault database. If combined with `NEW`, this modifier associates the password with the part name. A file password can be eight or fewer alphanumeric characters.

```
[NEW]
```

Enables you to put a new part in the database. Do *not* use this modifier if you entered `GET PART NEW` for this part. The dependent modifiers `PUBLIC`, `PRIVATE`, and `PROJECT` require the use of the `NEW` modifier.

[PUBLIC]

A classification group, allowing any user with the proper authority to access this part. This is the default if you do not specify a classification group.

[PRIVATE]

A classification group, allowing only your user name and the system administrator to have access to this part.

[PROJECT projectID]

A classification group, allowing only users assigned to that project identifier (ID) with the proper authority to have access to this part. The project identifier must already exist in Vault.

### Example PUT PART TEST.PART (CR)

```
CDMSON016I Sign on to EDM server NODENAME completed successfully.  
You have 0 EDM message(s).  
CDMREP882I The part has been replaced.  
CDMSOF017I Sign off from EDM completed successfully.  
TEST.PART  
    PUT into the database.
```

The messages in this example indicate that the part has been successfully stored in Vault and is no longer signed out to you.

If you attempt to return the part to the Vault database but the system cannot service your request, wait a minute or two and make your request again.

## UPDATE PART

Use the command `UPDATE PART` when you want to replace a part in Vault with your modified version but still keep the part available for further modification. If you have modify access to the part, you can use this command. Otherwise, a system message reminds you that you do not have modify-access to the named part. Use the command `PUT PART` when adding new drawings to a part.

### Valid Modes

Inside CADD5 5i, in or out of a Part or Drawing, in Model or Draw mode.

### Format

```
UPDATE PART partname NODE nodename [FILEPASSWORD password]
```

### Modifiers

```
[DMPASSWORD password]
```

Signs you on to the Vault node you are getting the file(s) from. You need only enter this `DMPASSWORD` once during a CADD5 5i session. Alternatively, you can include the following line in your `.caddsrc` file:

```
setenv DMPASSWORD password
```

`password` has an eight character maximum. You must reenter the password if you change nodes on a subsequent `GET` or `PUT` and the password on the new node differs from that of the previous node.

```
[PARTNAME] partname
```

Names the part exactly as you accessed it using the `GET PART` command.

```
[NODE nodename]
```

Names the requested node; the node that you specify is where the part exists in Vault. Supply both the modifier (`NODE`) and the actual node name. Node names can be as many as 18 alphanumeric characters.

Not required if you are using the default node.

FILEPASSWORD password

Is the password associated with one or more files in the part. Must be specified if passwords exist.

### Example UPDATE PART TEST.PART

```
CDMSON016I Sign on to EDM server NODENAME completed successfully.  
You have 0 EDM message(s).  
CDMUPT887I The part has been updated.  
CDMSOF017I Sign off from EDM completed successfully.  
TEST.PART  
    UPDATED in the database.
```

In this example, changes made to the part are updated and stored in the Vault database. The part is still signed out to you and you can continue to modify it.

## GET FILE

The `GET FILE` command allows you to sign out a single file or a group of files from Vault. Possible outcomes from this command are as follows:

- Have all named file(s) available with modify access.
- Specify `READ` as a modifier in the command line for read-only access to the file.
- A system message reports no file found under that name.

Please note: Optegra version 4.0 does not support CADD5 5i catalog.

You must include a file name with the `GET FILE` command. The password once used with a file is required for any `GET/PUT` operation on that file.

`GET FILE` is not recommended for any part file transfers. Use `GET PART` to ensure that all part related files are transferred.

Please note: Press `ESCAPE + Q` to interrupt multiple file transfers.

### Valid Modes

Within CADD5 5i, in or out of a Part or Drawing, and in Model or Draw mode.

### Format

```
GET FILE filename [modifiers]
```

### Modifiers

```
[FILENAME] filename
```

Names the file or files to be accessed. The modifier `FILENAME` is optional. The actual file name is required. If not used with `SINGLE`, the system assumes the request is for a CADD5 5i catalog.

```
[DMPASSWORD password]
```

Signs you on to the Vault node you are getting the file(s) from. You need only enter this `DMPASSWORD` once during a CADD5 5i session. Alternatively, you can include the following line in your `.caddsrc` file:

```
setenv DMPASSWORD password
```

`password` has an eight character maximum. You must reenter the password if you

change nodes on a subsequent GET or PUT and the password on the new node differs from that of the previous node.

[NODE nodename]

Names the requested node; the node that you specify is where the GET command goes to retrieve the data. Supply both the modifier (NODE) and the actual node name. Node names can be as many as 18 alphanumeric characters. Not required if you are using the default node.

FILEPASSWORD password

Is the password associated with the file or with one or more files in the CADD5 5i catalog. Must be specified if passwords exist. A file password can be eight or fewer alphanumeric characters. If you request a CADD5 5i catalog, all files with the entered password or no password are transferred.

[SINGLE]

Instructs the system to get only a single file, not an entire CADD5 5i catalog of files.

[READ prefix]

Indicates you want read-only access to the named file(s). The system attaches the prefix, followed by a period (.), to the specified file name, creating a new file name. Files are not signed out of the Vault database when READ is used. Instead, a copy of the current Vault file is renamed and sent to the local disk.

[REVISION n]

Use this modifier with the modifier READ to tell Vault which revision of the file you want. All associated files with that revision level will be transferred. If you do not enter a revision level, files with the highest revision level, if released, are returned. CADD5 5i GET/PUT commands support only numeric revision codes.

## Example GET FILE AA.HELP

```
GET FILE AA.HELP
CDMSON016I Sign on to EDM server NODENAME completed successfully.
You have 0 EDM message(s).
CDMGET606I All of the file(s) requested have been processed.
CDMSOF017I Sign off from EDM completed successfully.
AA.HELP transferred from the database.
```

In this example, the file is signed out to you and you can modify it.

## PUT FILE

The `PUT FILE` command gives you the ability to move a file or files, accessed with a `GET FILE` command, from your local disk to Vault. The `NEW` modifier allows you to move a new file to the database. The `PUT FILE` command is not recommended for part file transfers. Use `PUT PART` to ensure that all part related files are transferred.

Please note: Optegra version 4.0 does not support CADD5 5i catalog.

Please note: Press `ESCAPE + Q` to interrupt multiple file transfers.

### Valid Modes

Within CADD5 5i, in or out of a Part or Drawing, and in Model or Draw mode.

### Format

```
PUT FILE filename [modifiers]
```

### Modifiers

```
[FILENAME] filename
```

Names the file or CADD5 5i catalog of files to be returned to Vault. If you do not use `SINGLE`, the system assumes you want to move an entire CADD5 5i catalog.

```
[DMPASSWORD password]
```

Signs you on to the Vault node you are getting the file(s) from. You need only enter this `DMPASSWORD` once during a CADD5 5i session. Alternatively, you can include the following line in your `.caddsrc` file:

```
setenv DMPASSWORD password
```

`password` has an eight character maximum. You must re-enter the password if you change nodes on a subsequent `GET` or `PUT` and the password on the new node differs from that of the previous node.

Not required if you are using the default node.

[NODE nodename]

Names the requested node; the node that you specify is where the PUT command goes to put the data. Supply both the modifier (NODE) and the actual node name. Node names can be as many as 18 alphanumeric characters.

Not required if you are using the default node.

FILEPASSWORD password

Is the password associated with the file or one or more files in the CADD5 5i catalog. Must be specified if passwords exist. A password is *not* required when adding a new file or CADD5 5i catalog to the Vault database. If combined with NEW, this modifier associates the password with the file or CADD5 5i catalog name. A file password can be eight or fewer alphanumeric characters.

[SINGLE]

This modifier notes that the file is a single file, not an entire CADD5 5i catalog of files.

[NEW]

Indicates that you want to store a new file in the database. If you do not specify NEW and a CADD5 5i catalog is being returned to the database, only files already accessed are returned.

If you perform a GET FILE on a CADD5 5i catalog and then you add new files to the CADD5 5i catalog while it is in your local storage area, do the following: 1. Run the PUT FILE command on the CADD5 5i catalog without the NEW modifier to replace the files that were originally in the CADD5 5i catalog. 2. Run the PUT FILE command again, this time with the NEW modifier, to store the newly added files to the CADD5 5i catalog in the Vault database.

The dependent modifiers PUBLIC, PRIVATE, and PROJECT require the use of the NEW modifier.

[PUBLIC]

A classification group, allowing *any user* with the proper authority to access the part. This is the default if you do not specify a classification group.

[PRIVATE]

A classification group, allowing only your user name access to the part.

[PROJECT projectID]

A classification group, allowing only users assigned to that project identifier with the proper authority to access the part.

### Example PUT FILE HELP NEW PUBLIC SINGLE

```
CDMSON016I Sign on to EDM server NODENAME completed
successfully. You have 0 EDM message(s).
CDMSTR177I The file has been stored.
CDMSOF017I Sign off from EDM completed successfully.
HELP
    PUT into the database.
```

This example shows how to store a new file in Vault. Vault deletes your local copy. The **SINGLE** modifier indicates a single file only, not a CADD5 5i catalog of files.

### Example PUT FILE AA.HELP NEW PUBLIC

```
CDMSON016I Sign on to EDM server NODENAME completed
successfully. You have 0 EDM message(s).
CDMSTR177I The catalog has been stored.
CDMSOF017I Sign off from EDM completed successfully.
AA.HELP
    PUT into the database.
```

This example demonstrates how to return a CADD5 5i catalog of files to the database. Notice the **SINGLE** modifier is not used in this example.

# GET FILESET

This command allows you to transfer sets of related files, or *file sets*, from a Vault node to your own CADD5Station. The file sets are placed at CADD5 5i level in your local CADD5 5i create directory. You receive a confirmation message upon the successful transfer of each file in the file set. If the transfer cannot be completed, you receive an explanatory message, and the transfer is aborted.

## Valid Modes

Only at CADD5 5i level.

## Format

```
GET FILESET FILESETNAME filesetname [modifiers]
```

## Modifiers

```
FILESETNAME filesetname
```

Names the file set to be transferred from Vault to CADD5 5i. Must be specified.

```
[DMPASSWORD password]
```

Signs you on to the Vault node you are getting the file(s) from. You need only enter this DMPASSWORD once during a CADD5 5i session. Alternatively, you can include the following line in your `.caddsrc` file:

```
setenv DMPASSWORD password
```

`password` has an eight character maximum. You must re-enter the password if you change nodes on a subsequent GET or PUT and the password on the new node differs from that of the previous node.

```
[NODE nodename]
```

Indicates the node from which the file set is to be transferred. Optional.

```
FILEPASSWORD password
```

Is the password associated with one or more files in the file set. Must be specified if passwords exist.

[READ prefix]

Specifies that you do not want to have Vault sign out the file set. Names the prefix you want prepended to the file set name. Optional.

### Example GET FILESET TESTFS3 READ AB

```
CDMSON016I Sign on to EDM server NODENAME completed  
successfully. You have 0 EDM message(s).  
CDMREA606I All of the file(s) requested have been processed.  
CDMSOF017I Sign off from EDM completed successfully.  
TESTFS3  
    transferred from the database.
```

In this example, the **READ** modifier gives you a copy of the files in the file set **TESTFS3**, places the files in a subdirectory called **AB**, and allows you read-only access to the files in the database. The files are not signed out to you.

### Example GET FILESET TESTFS3

```
CDMSON016I Sign on to EDM server NODENAME completed  
successfully. You have 0 EDM message(s).  
CDMGET606I All of the file(s) requested have been processed.  
CDMSOF017I Sign off from EDM completed successfully.  
TESTFS3 transferred from the database.
```

## PUT FILESET

This command allows you to transfer sets of related files, or *file sets*, from your CADD5Station to a Vault node. File sets are placed in whichever node you specify in the command line. You must enter `PUT FILESET` from a CADD5Station at CADD5 5i level.

Only those file sets you have signed out with the `GET FILESET` command can be transferred. All files in the file set must already exist in the target Vault database.

You receive a confirmation message upon the successful transfer of each file in the file set. Once a file set is successfully transferred, all files are deleted from the CADD5Station directory, including the cross-reference file.

### Valid Modes

Only at CADD5 5i level.

### Format

```
PUT FILESET FILESETNAME filesetname [modifiers]
```

### Modifiers

```
FILESETNAME filesetname
```

Names the file set to be transferred from the CADD5 5i user to Vault. Must be specified.

```
[DMPASSWORD password]
```

Signs you on to the Vault node you are *putting* the file set to. You need only enter this `DMPASSWORD` once during a CADD5 5i session. Alternatively, you can include the following line in your `.caddsrc` file:

```
setenv DMPASSWORD password
```

`password` has an eight character maximum. You must re-enter the password if you change nodes on a subsequent `GET` or `PUT` and the password on the new node differs from that of the previous node.

[NODE nodename]

Names the node to which the file set is to be transferred. Must be the node the file set was retrieved from. Optional.

### Example PUT FILESET TESTFS3

```
CDMSON016I Sign on to EDM server NODENAME completed  
successfully. You have 0 EDM message(s).  
CDMREP882I The file set has been replaced.  
CDMSOF017I Sign off from EDM completed successfully.  
TESTFS3  
    PUT into the database.
```

This example stores TESTFS3 in Vault and deletes your local copies. The files are no longer signed out to you.

# Quick Reference for Vault Commands in CADD5 5i

## PUT PART

```
PUT PART  
[PARTNAME] partname  
NODE nodename  
DMPASSWORD password  
FILEPASSWORD password  
NEW  
PUBLIC  
PRIVATE  
PROJECT projectID
```

## GET PART

```
GET PART  
[PARTNAME] partname  
NODE nodename  
DMPASSWORD password  
FILEPASSWORD password  
READ local directory name  
REVISION x  
NEW  
PUBLIC  
PRIVATE  
PROJECT projectID
```

## UPDATE PART

```
UPDATE PART  
[PARTNAME] partname  
NODE nodename  
FILEPASSWORD password
```

## GET FILE

```
GET FILE  
[FILENAME] filename  
NODE nodename  
SINGLE  
DMPASSWORD password  
FILEPASSWORD password  
READ prefix  
REVISION n
```

## PUT FILE

PUT FILE  
[FILENAME] filename  
NODE nodename  
DMPASSWORD password  
FILEPASSWORD password  
SINGLE  
NEW  
PUBLIC  
PRIVATE  
PROJECT projectID

## GET FILESET

GET FILESET  
FILESETNAME filesetname  
DMPASSWORD password  
NODE nodename  
FILEPASSWORD password  
READ prefix

## PUT FILESET

PUT FILESET  
FILESETNAME filesetname  
DMPASSWORD password  
NODE nodename

# Corresponding Commands

---

This appendix correlates the Vault functionality provided by the command line with that available on the Graphical User Interface.

- Command Line and GUI Functionality

# Command Line and GUI Functionality

Command	View Menu Selection	Command Menu/Button Selection
addag	View >Admin>Authority Groups	File>New
addaset	View>Admin>Attribute Sets	File>New
addattr	View>Admin>Attributes	File>New
addcl	View>Admin>Command Lists	File>New
addevals	View>Admin>Enumerations	Edit>Enumeration; Add
addfs	View>Vault>File Sets	File>New
addmas	View>Admin>Attribute Sets	Edit>Add Member
addmfs	View>Vault>File Sets	Edit>Members; Add
addmul	View>Admin>User Lists	Edit>Members; Add
addobj	View>Vault>Binders (View>Vault>Binders; File>Explode)	File>New
addp	View>Admin>Projects	File>New
addrel	View>Vault>Binders (View>Vault>Binders; File>Explode; select a binder)	Edit>Binder; Add
addrs	View>Admin>Revision Sequences	File>New
addrule	View>Admin>Attribute Sets	Edit>Add Rule
adds	View>Admin>Projects	Edit>Status Levels; Add
addsub - Distributed File	View>Distributed>Files	Administration>Distribute On Event; Create
addsub - Distributed Part	View>Distributed>Parts	Administration>Distribute On Event; Create
addsub - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file)	Administration>Notify On Event; Create
addsub - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part)	Administration>Notify On Event; Create
addu	View>Admin>Users	File>New
addul	View>Admin>User Lists	File>New
addup	View>Admin>Projects	Edit>Users; Add
addusa	View>Admin>Projects	Edit>Status Levels; Exit Criteria; Create
advlist	View>Admin>Vault Lists	File>New
advlmem	View>Admin>Vault Lists	Edit>Members (Depends on Candidate and Member Lists)
admcopy - Authority Group	All views	Administration>Copy>Authority Group
admcopy - Command List	All views	Administration>Copy>Command List

<b>Command</b>	<b>View Menu Selection</b>	<b>Command Menu/Button Selection</b>
admcopy - User List	All views	Administration>Copy>User List
associat - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Edit>User Attributes
associat - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part) (View>Distributed>Parts)	Edit>User Attributes
chgag	View>Admin>Authority Groups	Edit>Authority Group
chgatrb	View>Admin>Type Definitions	Edit>Type Definition; "Attributes" Edit
chgatrbs - Binder	View>Vault>Binders (View>Vault>Binders; File>Explode; select a binder)	Edit>Class Attributes
chgatrbs - Instance/Relationship	View>Vault>Binders (View>Vault>Binders; File>Explode; select a binder)	Edit>Binder; Edit
chgattr - Numeric Attribute	View>Admin>Attributes	Edit>Unit of Measure
chgattr - All Attributes	View>Admin>Attributes	Edit>Protection
chgcl	View>Admin>Command Lists	Edit>Command List
chgctl	View>Admin>Command triggers	Edit>Command Trigger
chgdesc - Attribute Group	View>Admin>Type Definitions	Edit>Type Definition; "Attribute Groups" Edit Description
chgdesc - Binder	View>Vault>Binders	Edit>Description
chgdesc - Enumeration	View>Admin>Enumerations	Edit>Description
chgdesc - /Instance/Relationship	View>Vault>Binders	Edit>Binder; Edit Description
chgdesc - Type Definition	View>Admin>Type Definitions	Edit>Description
chgfa - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Edit>File Attributes
chgfa - File Set	View>Vault>File Sets	Edit>File Attributes
chgfa - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part) (View>Distributed>Parts)	Edit>File Attributes
chgfc1 - Binder	View>Vault>Binders (View>Vault>Binders; File>Explode; select a binder)	Edit>Classification
chgfc1 - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Edit>Classification
chgfc1 - File Set	View>Vault>File Sets	Edit>Classification

<b>Command</b>	<b>View Menu Selection</b>	<b>Command Menu/Button Selection</b>
chgfcl - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part) (View>Distributed>Parts)	Edit>Classification
chgfpw - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Edit>Password
chgfpw - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part) (View>Distributed>Parts)	Edit>Password
chgfrev - Binder	View>Vault>Binders (View>Vault>Binder; File>Explode; select a binder)	Edit>Revision
chgfrev - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Edit>Revision
chgfrev - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part) (View>Distributed>Parts)	Edit>Revision
CHGFSC - Binder	View>Vault>Binders (View>Vault>Binders; File>Explode; select a binder)	Edit>File Status Level
chgfsc - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Edit>File Status Level
chgfsc - File Set	View>Vault>File Sets	Edit>File Status Level
chgfsc - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part) (View>Distributed>Parts)	Edit>File Status Level
chgfsp	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Edit>Storage Pool
chgmas	View>Admin>Attribute Sets	Edit>Members
chgp	View>Admin>Projects	Edit>Project
chgs	View>Admin>Projects	Edit>Status Levels; Edit
chgsps - From Admi View	View>Admin>Storage Pools	Edit>Storage Pool Status
chgsps - From Files View	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Edit>Storage Pool; "Storage Pool"; Edit
chgu	View>Admin>Users	Edit>User
chgup	View>Admin>Projects	Edit>Users; Edit
chgupw	View>Admin>Users	Edit>Password
copy - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Vault>Duplicate

<b>Command</b>	<b>View Menu Selection</b>	<b>Command Menu/Button Selection</b>
copy - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part) (View>Distributed>Parts)	Vault>Duplicate
decl	View>Admin>Command Lists	Edit>Delete
defagr	View>Admin>Type Definitions	Edit>Type Definition; "Attribute Groups" Create
defattr	View>Admin>Type Definitions	Edit>Type Definition; "Attributes" Create
defenum	View>Admin>Enumerations	File>New
defotype	View>Admin>Type Definitions	File>New
delete	All views	Administration>Delete Marked Files
delevals	View>Admin>Enumerations	Edit>Enumeration; Delete
delenum	View>Admin>Enumerations	Edit>Delete
delag	View>Admin>Authority Groups	Edit>Delete
delagr	View>Admin>Type Definitions	Edit>Type Definition; "Attribute Groups" Delete button
delaset	View>Admin>Attribute Sets	Edit>Delete
delattr	View>Admin>Type Definitions	Edit>Type Definition; "Attributes" Delete
delattr	View>Admin>Attributes	Edit>Delete
dello - All Before	View>Admin>Log Entries	Edit>Delete All Before
dello - All Entries	View>Admin>Log Entries	Edit>Delete All Entries
dello - All Since	View>Admin>Log Entries	Edit>Delete All Since
dello - Within Range	View>Admin>Log Entries	Edit>Delete
delobj	View>Vault>Binders (View>Vault>Binders File>Explode; select a binder)	Edit>Delete
delotype	View>Admin>Type Definitions	Edit>Delete
delov	All views	Administration>Delete Old Versions
delp	View>Admin>Projects	Edit>Delete
delrel	View>Vault>Binders (View>Vault>Binders; File>Explode; select a binder)	Edit>Binder; Delete
delrs	View>Admin>Revision Sequences	Edit>Delete
delrule	View>Admin>Attribute Sets	Edit>Delete Rule
dels	View>Admin>Projects	Edit>Status Levels; Remove
delsub - Distributed File	View>Distributed>Files	Administration>Distribute On Event; Delete
delsub - Distributed Part	View>Distributed>Parts	Administration>Distribute On Event; Delete
delsub - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file)	Administration>Notify On Event; Delete

<b>Command</b>	<b>View Menu Selection</b>	<b>Command Menu/Button Selection</b>
delsub - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part)	Administration>Notify On Event; Delete
delu	View>Admin>Users	Edit>Delete
delul	View>Admin>User Lists	Edit>Delete
export - File	View>Distributed>Files	Administration>Distribute Now
export - Part	View>Distributed>Parts	Administration>Distribute Now
get - Binder	View>Vault>Binders (View>Vault>Binders; File>Explode; select a binder)	Vault>Check Out>Lock
get - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Vault>Check Out>Lock
get - File Set	View>Vault>File Sets	Vault>Check Out>Lock
get - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part) (View>Distributed>Parts)	Vault>Check Out>Lock
purge -File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Administration>Purge>Selected Object
purge - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part) (View>Distributed>Parts)	Administration>Purge>Selected Object
read - Binder	View>Vault>Binders (View>Vault>Binders; File>Explode; select a binder)	Vault>Check Out>Do Not Lock
read - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Vault>Check Out>Do Not Lock
read - File Set	View>Vault>File Sets	Vault>Check Out>Do Not Lock
read - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part) (View>Distributed>Parts)	Vault>Check Out>Do Not Lock
readattr - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Edit>User Attributes
readattr - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part) (View>Distributed>Parts)	Edit>User Attributes
readmsg	All views	Messages>Read
register - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file)	Administration>Register
register - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part)	Administration>Register
remfs	View>Vault>File Sets	Edit>Delete
remmas	View>Admin>Attribute Sets	Edit>Delete Member
remmfs	View>Vault>File Sets	Edit>Members; Remove

<b>Command</b>	<b>View Menu Selection</b>	<b>Command Menu/Button Selection</b>
remmul	View>Admin>User Lists	Edit>Members; Remove
remup	View>Admin>Projects	Edit>Users; Remove
remusa	View>Admin>Projects	Edit>Status Levels; Exit Criteria; Remove
remvlist	View>Admin>Vault Lists	Edit>Delete
remvlmem	View>Admin>Vault Lists	Edit>Members (Depends on Candidate and Member Lists)
replace - Binder	View>Vault>Binders (View>Vault>Binders; File>Explode; select a binder)	Vault>Check In>Unlock
replace - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Vault>Check In>Unlock
replace - File Set	View>Vault>File Sets	Vault>Check In>Unlock
replace - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part) (View>Distributed>Parts)	Vault>Check In>Unlock
reqrvw - Binder	View>Vault>Binders (View>Vault>Binders; File>Explode; select a binder)	Review>Request
reqrvw - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Review>Request
reqrvw - File Set	View>Vault>File Sets	Review>Request
reqrvw - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part) (View>Distributed>Parts)	Review>Request
reserve	View>Local>Files	Vault>Reserve
reset - Binder	View>Vault>Binders (View>Vault>Binders; File>Explode; select a binder)	Vault>Unlock
reset - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Vault>Unlock
reset - File Set	View>Vault>File Sets	Vault>Unlock
reset - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part) (View>Distributed>Parts)	Vault>Unlock
rsvp - Binder	View>Vault>Binders (View>Vault>Binders; File>Explode; select a binder)	Review>Respond
rsvp - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Review>Respond
rsvp - File Set	View>Vault>File Sets	Review>Respond
rsvp - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part) (View>Distributed>Parts)	Review>Respond
sendmsg	All views	Messages>Send

<b>Command</b>	<b>View Menu Selection</b>	<b>Command Menu/Button Selection</b>
signoff	All views	File>sign Off
signon	All views	File>Sign On
signout - Binder	View>Vault>Binders (View>Vault>Binders; File>Explode; select a binder)	Vault>Check Out>Lock Only
signout - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Vault>Check Out>Lock Only
signout - File Set	View>Vault>File Sets	Vault>Check Out>Lock Only
signout - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part) (View>Distributed>Parts)	Vault>Check Out>Lock Only
store -Directory	View>Local>Directories	Vault>Store
store - File	View>Local>Files	Vault>Store
store - Part	View>Local>Parts	Vault>Store
unmark - All Vault Files	All views	Administration>Unmark>All Vault Files
unmark - Selected Object	All views	Administration>Unmark>Selected Object
update - Binder	View>Vault>Binders (View>Vault>Binders; File>Explode; select a binder)	Vault>Check In>Keep Lock
update - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Vault>Check In>Keep Lock
update - File Set	View>Vault>File Sets	Vault>Check In>Keep Lock
update - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part) (View>Distributed>Parts)	Vault>Check In>Keep Lock
veruattr - File	View>Vault>Files (View>Vault>Binders; File>Explode; select a file) (View>Distributed>Files)	Edit>User Attributes
veruattr - Part	View>Vault>Parts (View>Vault>Binders; File>Explode; select a part) (View>Distributed>Parts)	Edit>User Attributes

---

# Glossary

---

The Glossary contains terms for end users, managers who oversee Vault operations, and system administrators.

access authority

Authority associated with your Vault User ID determines which Vault files and parts you can access.

action

A function to perform on behalf of a subscription. Examples are send message, export, register. Used by the Distributed Vault product.

Action Manager

The Vault service that processes actions requested by the Event Manager and by the Register function. Part of the Distributed Vault product.

Administrative Server (DN)

The Vault service that oversees administrative functions that do not impact files or parts, such as adding a user or creating a file set. See Vault server.

AE

See Application Entity (AE)

aggregation

A dependent relationship between a parent binder and the files and binders in that binder. In an aggregation relationship, a change to one member of the collection of items in the binder affects the other members. You use aggregation when you want constraints and dependencies between

members. All aggregated members have the same life cycle as their parent binder. A binder member may be associated to multiple binders but can only be aggregated to one binder. See *association*.

Application Entity (AE)

A Vault server process, such as the Administrative Server, Attribute Manager, Data Distribution Server, Data Management Facility, and Log Services Manager. Abbreviated as AE. See Vault server.

archived file

A file that has been moved to tape storage and is no longer available in the vault storage pool. The metadata associated with archived files enables them to be restored for active use.

assembly

A group of parts, and their interrelationships, that comprise a partial or complete product.

association

An independent relationship between a parent binder and the files and binders in that parent binder. All members can have different life cycles. A binder member may be associated to more than one binder. You can use the association relationship to manage a group of objects or a large project with pieces managed by different groups. See *aggregation*.

attribute

Information about a Vault object, that is, a file or part (group of files). A Vault object has built-in attributes such as the file's physical location, classification, revision code, status level, and version. The system administrator can also define custom attributes. Attribute data for a file is stored in the RDBMS, whereas the file itself is stored in a storage pool.

attribute group

A collection of binder/relation attributes. Attribute groups are associated to binders and relations by a `typedef`.

Attribute Manager

The part of the Vault server that maintains the user-defined attribute data on parts and files.

## attribute set

A collection of user-defined file and part attributes. Attribute sets are associated with files and parts through rules.

## audit file

A local file that stores results generated by certain Vault commands. For example, when you mark files or read files from disk or tape, that information is recorded in the audit file.

## audit log

The record of all transactions for a Vault, both successful and failed.

## authority group

A named set of authority numbers. Groups are created by a Vault administrator.

## authority number

A two-digit integer (00-99) that controls access to files by users as part of an authority scheme. Users can access files with numbers less than or equal to their authority numbers. Authority numbers are assigned by an administrator (public files) or by a project administrator (project files). A given file is assigned an authority number through its status level.

## authority scheme

A set of authority levels for each status level assigned to users. An authority scheme determines who can access a particular file at a particular point in its design cycle. Each user has a public authority scheme for public files and a project authority scheme for each project. A project may use a job type for an authority scheme.

## base class

Name of a predefined class of objects from which the object type (binder, relationship, file, or part) is derived.

## binder

A user-defined collection of Vault files, parts, or other binders. Binders, unlike file sets, have ownership, revision and status level. They may also have user-defined attributes. After you create a binder, you add objects by relating them to the binder. Binder members can belong to more than one binder. See also relationship.

Data Management Facility (DMF)

Part of the Data Distribution Service that determines if a transfer of a file may be performed. Abbreviated as DMF. See Vault server.

data type

A definition of a set of data that specifies the possible range of values of the set, the operations that can be performed on the values, and the way in which the values are stored in memory. Data types are string, real, integer, and enumeration.

design cycle

The sequence of phases that a file or part passes through until the revision is complete (released). In Vault, a phase is known as a status level. In the Vault, phases are grouped into a status level partition. The public authority scheme and each project has its own design cycle.

directory

A group of unrelated files in the local file system. All files in a local directory can be stored (checked into Vault) As a group, though no association remains among them after they are stored.

distributed parts and files

Parts and files that have been copied to Vaults other than the one in which they were created. Possible states in the new Vault include: replicated for read, replicated for write. Used by the Distributed Vault product.

Distributed Object Directory (DOD)

A repository that contains Vault locations of all registered files and parts. Part of the Distributed Vault product. A Vault in a Distributed Vault environment consults the DOD to locate objects that you request.

Distributed Vault

Distributed Vault (DV) provides for a transparent and cooperative use of Vaults in a distributed environment. This arrangement allows you to access registered files and parts on any of the configured Vaults. You do not need to know the locations of these items in the Distributed Vault.

DMF

See Data Management Facility (DMF)

DN

See Administrative Server (DN)

DOD

See Distributed Object Directory (DOD)

domain

A name which can be used to define all of the related client and server processes which constitute a single instance of an Optegra Vault environment. All Optegra Vault client and server processes execute within the context of a single domain. DOMAIN is synonymous with VAULT ID.

DV

See Distributed Vault

EDM

Enterprise Data Manager

enumerated list

An exhaustive set of values to allow for a binder/relation attribute.  
Example: Color\_list: Red, Green, Blue.

Enumeration

A value set, or collection of strings under user control. If an attribute is associated with an enumeration, it means that the value of the attribute is one of the strings of the collection.

event

A data base change such as object status change, object revision change, object metadata change. The subscription command enables events to be monitored so that when they occur, actions can be performed automatically.

Event Manager

The Vault service that monitors events that have been logged by event triggers. Part of the Distributed Vault product.

event trigger

An Oracle procedure that watches for events that have subscriptions and logs them to an event queue database table. Part of the Distributed Vault product.

export services

Export services replicate registered files and parts from one vault (the exporter) to other vaults (the importers) in the Distributed Vault environment.

file

The smallest unit of named data. In the Vault, a file has associated metadata, or attributes, such as its owner, history and status. These attributes protect and manage the file. Files may be grouped into parts, file sets, or binders.

file set

A user-defined group of Vault files, parts, or other file sets. Members can belong to more than one file set. Unlike a part, a file set does not have attributes. Some vault commands allow you to operate on all members of a file set at one time.

host

A host is a computer. It is represented by its name, called the host name. When configured with Optegra Vault, a host can be referred to as a Vault host.

import services

Import services replicate registered files and parts from an exporter Vault to importing Vaults in the Distributed Vault environment.

instance attribute

Attributes on relations are called instance attributes.

instance

A parent-child relation that occurs more than once.

## in-work

The designation for a file or part that is currently available for modification. Only the highest revision of a part or file may be inwork.

## Internet Protocol (IP)

Part of TCP/IP. One part of the protocol is a computer addressing scheme. IP addresses use 32-bits to identify a node. They are usually represented in dotted decimal notation (for example, 102.52.94.97) with one part being your network ID and the other your system ID.

## Interactive Query Facility (IQF)

A querying capability that enables you to extract and output information from the Vault RDBMS about files, projects, authority schemes, and so forth.

## IP

See Internet Protocol (IP)

## IQF

See Interactive Query Facility (IQF)

## job type

A set of privileges in a project authority scheme created by a project administrator. It consists of a command list and an authority group. Each project has one or more job types. Each user assigned to a project has a single job type on that project.

## life cycle

The set of all design cycles.

## Log Services Manager

The Vault process that receives messages from each command and stores them optionally into the database audit log table.

## mailing list

All the Vault user IDs on a user list. Users on the list are notified of a review and are informed of the results. Vault uses the user list to send action messages or notification messages for reviews. User lists are created by a Vault project administrator.

message

A Vault-maintained electronic message, usually associated with reviews but also associated with the outcome of subscriptions.

metadata

Information about a Vault file such as its classification, location, revision, status level, version, and so forth. This data is stored in the RDBMS, whereas the data files themselves are stored in a storage pool.

network

A connection among two or more computers that enables them to directly transfer information. The Vault uses the TCP/IP network protocol.

Network Services Manager (NSM)

NSM performs system and network management functions for nodes that access the Vault. The Process Manager is an example of an application under the control of NSM. See Vault server.

node

In NSM terms, a node represents the name of a computer upon which Vault client OR server processes are executed. Generally, a node is identical to a host, although the term 'host' also has connotations relative to internet work addressing, whereas a node name has no such relationship.

notification message

A message sent by the Vault to inform you of the start of a review cycle.

Notification Subscription

A list of users created by the Vault Administrator or the Project Leader. The users on the list will be notified when certain events take place during a project.

NSM

See Network Services Manager (NSM)

Oracle

A relational database management system marketed by Oracle Corporation. Vault applications use the Oracle version supported by Optegra for basic database management functions.

**owner**

The user or project to which a file or part belongs. According to the classification scheme, public files have no owner, whereas private files are owned by a Vault user and project files are owned by a project. Private files are accessible only by the user that owns them. Project files are accessible by users with authority on that project.

**part**

A group of related files from within an application, for example, CADDs. Part members are controlled together and usually share the same name prefix. A file can belong to only one part.

Each part has a classification, revision, status, and owner (public, project, or private) that it shares with its constituent files. Part types are defined with the environment parameter (GUI term is Local Rule) when a file is first stored in Vault.

**placeholder file**

A reserved Vault file name. This name holds a place for a future file. The reserved file name contains only metadata and no corresponding data file.

**PRI**

See Private (PRI)

**Private (PRI)**

The classification assigned to files and parts that are owned by a single user ID.

**PRO**

See Project (PRO)

**Process Manager**

The program responsible for managing Vault process resources. It:

- [a] Coordinates the Vault's startup and shutdown sequence
- [b] Validates all incoming client AND server service requests.
- [c] Allocates servers to clients, when requested, and starts additional servers, in response to these requests, if necessary and possible.
- [d] Monitors all running client and server processes

The Process Manager program generally resides on the same machine as Vault's server processes, although this is not a requirement.

Product structure

The diagram that represents all the relationships between the parts and subassemblies within an assembly.

Project (PRO)

The classification assigned to files and parts owned by a project. Access to project files is controlled by the project's authority scheme.

PUB

See Public (PUB)

Public (PUB)

The file classification assigned to Vault files and parts that are accessible to every user depending on their authority within the public authority scheme.

restore

To return a file/part to active use from archived status.

RDBMS

Relational Database Management System. Oracle and SQL/DS are RDBMS products used with Vault applications.

read authority

A level of permissions giving you the ability to read but not modify a file. The Vault option for reading copies a Vault file to your computer for viewing. This operation does not allow you to modify the file.

registration

A process by which you register files to the Distributed Object Directory for access on Distributed Vaults. A registered item can be accessed by users on other Vaults in the Distributed Vault environment.

relationship

The connector between a parent binder, or container, and the members within it: files, parts, or other binders. Relationships can be an aggregation with a dependency on other objects or an association independent of other objects. See association and aggregation.

**released**

A designation for a file or part that is no longer being modified. Such objects are in a released status level partition. When you request to modify a released part or file, a new revision of the object is created at the initial inwork status level.

**review definition**

The definition of the criteria for the approval of a file set by the Vault Administrator or Project Leader. A review definition can be unanimous, majority or based on the number of rejects.

**review list**

A user list whose members are asked to review a portion of work.

**review message**

A message sent by the Vault to ask you to review a portion of work.

**revision sequence**

A set of revision codes predefined for use in projects. Newly created files and parts are assigned the first revision code in their project's revision sequence. When a released object is checked out, a new object with the next revision code in the sequence is created.

Some projects do not have assigned revision sequences. The objects in such projects must be manually assigned a revision code before they are released.

**revision code**

An alphanumeric designation which, together with a name, uniquely identifies a part or file. A file or part may exist in Vault under more than one revision code, but only the latest revision can be in-work. All other revisions are released. Parts and files in projects without an assigned revision sequence are created with a blank revision code which must be changed to a non-blank value before the object is released.

**root binder**

The binder at the top of the hierarchy of binders.

rulebase

A set of rules to define and handle a group of files as a single part within an application environment such as CADDSS. User-defined rulebases are implemented within the Vault as an application or executable file.

selection scope

A parameter that allows you to specify the type of entity you want to store or sign out. Valid entities are file (F), part (P), file set (S), or list (L).

SQL

See Structured Query Language (SQL)

status level

An alphanumeric code identifying the current phase of work on the Vault file or part. Status levels are a subdivision of the status partition levels: Pre-inwork, In-work, or Released. Each status code is associated to an authority number. The Vault compares the authority number of the file's status code with the authority number of a user to determine whether or not a given user can access a given file.

status level partition

A set of three design cycle phases to control whether you can modify a file: Dormant (optional), In-work (I), and Released (R). Status partitions are further divided into status levels. The In-work and Released partitions must have at least one assigned status level.

storage pool

The Vault term for the location of user files under its control. The Vault server requires at least two and preferably seven or more storage pools for each Vault installation.

Structured Query Language (SQL)

An ANSI standard for retrieving and manipulating data stored in a relational database management system.

subscription

A user-defined association between an object (part/file/project), an event on that object, and an action to take as a result of the event. Available with the Distributed Vault product.

## tablespace

Logical storage units within Vault for the relational database. The usable data, such as tables and indexes, is stored in tablespaces. Tablespaces are physically stored in one or more data files.

## TCP/IP

Transmission Control Protocol/Internet Protocol. An Ethernet-based local area network communication standard.

## type definition

A user definition of the attributes and attribute groups of the objects to be included within a binder. Before binders can be created, a `typedef` must be defined for them.

## typedef

See type definition.

## unit of measure

An optional attribute of a numeric part or file attribute. Unit of measure specifies the type of quantity represented by numbers, for example, inches, feet, or miles.

## User defined Table

An Oracle database table created outside of the Optegra Vault that can be placed under Vault access and security control.

## User ID

An account set up for a Vault user.

## user list

A list of Vault user IDs for use with a particular file or project. Vault uses the user list to notify users of a review and inform them of the results. User lists are created by the Vault project administrator.

## Vault host

The the name of the computer on which the server processes of a vault DOMAIN run. Ordinarily, the name of the DOMAIN and the name of the Vault host are the same, except that the DOMAIN name is the upper-cased value of the Vault host.

Vault list

A Vault list is a list of DOMAINS, that is, a list of Vault IDs. Vault lists are used by Distributed Vault, for registering and exporting objects. A Vault list determines which Vault IDs (DOMAINS) are to receive (Import) items from an exporting Vault.

Vault server

Synonymous with Vault host.

version

A Vault object is uniquely identified by its name and revision. As files are checked in and out of Vault, multiple copies of the same name and revision are created in Vault. These multiple copies are distinguished internally by their version number. In general, only the latest version (version 0) of an object is accessible to users, but old versions are manipulated by Vault administrative tasks such as backup and recovery. Vault uses a time stamp to identify old versions.

write authority

A level of permissions giving you the ability to modify a file.

---

# Index

---

## Symbols

.attr file 4-6  
.Edmgui file A-2

## Numerics

00 file type 4-6

## A

Access authority 2-2  
Accessibility, restricting for objects (Mark command) 4-17  
Accessing Vault 2-3  
    CADD5 5i, using B-1  
    command line 2-4  
Add Member dialog box 6-29  
addag command C-2  
addaset command C-2  
addattr command C-2  
addcl command C-2  
addevals command C-2  
addfs command C-2  
adding members to a binder 6-29  
addmas command C-2  
addmfs command C-2  
addmul command C-2  
addobj command C-2  
addp command C-2  
addrel command C-2  
addrs command C-2  
addrule command C-2  
adds command C-2  
addsub command C-2  
addu command C-2  
addul command C-2  
addup command C-2  
addusa command C-2  
addvlist command C-2  
addvlmem command C-2  
admcopy command C-2  
Administrator settings A-2  
Aggregation (type of relation) 6-12  
Approval schemes, review 2-17  
Archiving objects (Mark command) 4-17  
associat command C-3  
Association (type of relation) 6-11  
Attribute groups, defining for binders 6-5  
    naming 6-19  
Attributes 3-11  
    .attr files (CADD5 parts) 4-6  
    binder 6-5  
        defining 6-18  
    changing 3-12  
    editing 6-31  
    file (user defined) 3-12  
    instances, creating for 6-32  
    mandatory 6-7  
        defining 6-20  
    optional 6-9  
    relations 6-6  
    user defined 3-12  
Audit file 4-15  
    reserved file name, indicating 5-5  
Authority schemes 3-9  
    objects, storing 2-6

- projects 3-9
- review cycle 2-16
- Authority, Vault access 2-2

## B

- Base class (binders) 6-4
- Base Directory dialog box 2-7
- Binder Members dialog box 6-29
- Binders 6-2
  - aggregation 6-12
  - association 6-11
  - attribute groups, defining 6-5
    - naming 6-19
  - attributes 6-5
    - editing 6-31
  - base class 6-4
  - checking in
    - procedure 6-36
  - checking out 6-14
    - procedure 6-35
  - class attributes, creating 6-27
  - deleting 6-37
  - example (illustration) 6-4
  - flowchart 6-17
  - hierarchy, viewing 6-33
  - introduction 3-3
  - members, adding 6-29
  - members, removing 6-34
  - object types, defining 6-18
  - operations 6-14
  - relations 6-3
    - types of 6-11
  - review 6-14
  - setting up 6-17
  - types 6-4
- binders
  - checking in 6-36
  - creating an enumeration 6-23
  - removing
    - members 6-34
  - setting up binder attribute types
    - defining attributes 6-21
    - defining data types 6-23
  - viewing
    - hierarchy 6-33
  - working with 6-25

## C

- CAD/CAM parts 3-2
  - storing in Vault 4-6
- CADDS parts 3-2
  - 00 file type 4-6
  - attribute file, Vault (.attr) 4-6
  - character conversions, Vault (file names) 4-5
  - checking out 4-9
  - file names (uppercase) 3-5
  - storing 4-4
    - UNIX names (checked out parts) 4-11
- Cascade option 6-37
- Case-sensitivity in UNIX (file naming conventions) 3-5
- Change Vault Attributes dialog box 3-12
  - object properties, changing 3-16
- Character conversions, Vault (CADDS file names) 4-5
- Check In command 2-9
- Check Out command 2-8
- Check Out dialog box
  - local directory, specifying 4-9
- Checked out objects, duplicating 4-15
- Checking in objects 2-11
  - binders
    - procedure 6-36
  - rules 4-13
  - rules for file sets 5-9
- checking out binders 6-35
- Checking out objects 2-9
  - binders 6-14
    - procedure 6-35
  - naming 4-9
  - rule for file sets 5-9
  - UNIX names 4-11
  - valid UNIX name (Local Name) 4-8
- Checkout dialog box 2-10
- chgag command C-3
- chgatrb command C-3
- chgatrbs command C-3
- chgattr command C-3
- chgcl command C-3
- chgctl command C-3
- chgdesc command C-3
- chgfa command C-3
- chgfc1 command C-3
- chgfpw command C-4
- chgfrev command C-4

- chgfscc command C-4
- chgfsfp command C-4
- chgmas command C-4
- chgp command C-4
- chgs command C-4
- chgspcs command C-4
- chgu command C-4
- chgup command C-4
- chgupw command C-4
- child and parent relations (binders) 6-3
- cisgnon command (command line login) 2-4
- Class attributes (binders) 6-5
  - creating 6-27
- Class, base (binders) 6-4
- Classes of objects 6-2
  - defining 6-20
- Classification 3-9
  - changing 3-17
- Colors, customizing in Vault A-4
- Command line, Vault C-1
- Commands (menu), table of 2-8
- Common login, using 2-4
  - command line considerations 2-4
- Comparison of Vault objects 3-4
- Concurrent Notification Subscription 2-17
- Connecting to Vault 2-2
- Connectors (relations) 6-3
- Contract deliverables, managing (binders) 6-2
- Control cycle, Release/Revision 2-16
- Conventions, naming 3-5
- Copies of objects (Duplicate command) 4-15
  - naming 4-16
- copy command C-4
- current working directory 4-9
- Custom Entry List A-3
- Custom menu (invoking other applications) A-3
- Custom parts (other software) 3-2
  - invoking from Vault (Custom menu) A-3
  - storing 4-6
- Customizing Vault GUI A-1
  - dialog boxes A-5
  - Edmgui resouce file A-2
  - executables, invoking (Custom menu) A-3
  - Preferences menu A-4
- CVPATH logical 4-9

## D

- Data types (attributes) 6-5
- Date (attribute data type) 6-5
- decl command C-5
- defagrp command C-5
- defatrb command C-5
- Default local directory, changing 2-7
- Defaults for duplicate object names 4-16
- defenum command C-5
- Define Attribute dialog box 6-18
- Define Attribute Group dialog box 6-19
- defotype command C-5
- delag command C-5
- delagrp command C-5
- delaset command C-5
- delatrb command C-5
- delattr command C-5
- delenum command C-5
- delete command C-5
- delevals command C-5
- Deliverables, managing (binders) 6-2
- dellog command C-5
- delobj command C-5
- delotype command C-5
- delov command C-5
- delp command C-5
- delrel command C-5
- delrs command C-5
- delrule command C-5
- dels command C-5
- delsub command C-5
- delu command C-6
- delul command C-6
- Design cycle 2-16
  - revision code, changing 3-16
- Design review folders (binders) 6-2
- Design Rule Customization box A-5
- Dialog boxes, customizing A-5
- Directory, Local
  - checked out objects, storing 4-9
- Directory, local 4-3
  - default, changing 2-7
  - specifying (Preferences) 2-10
- Distributed Files or Parts browser 2-21
- Distributed Vault 2-20
  - export operation 2-21
  - objects, distributing 2-21

- objects, registering 2-20
- registered objects, viewing 2-20
- Distributed Vault browser 2-20
- Distributed view, registering an object 2-20
  - marking (archive or delete) 4-17
- Distributing objects 2-21
- Do Not Lock option, Check Out command 2-9
- Documentation, printing from Portable Document Format (PDF) file xv
- Drawing objects, managing 6-2
- Duplicate command 2-9
  - stored objects 4-15
- Duplicating a file 2-11
  - revision code, resulting 3-8
- Duplicating objects
  - naming 4-16
- Duplicating stored objects 4-15

## E

- Edit Local Selection Criteria dialog box 2-7
- editing
  - binder class attributes 6-27
- Editor button 2-3
- EDM.DEFAULTS file 4-2
  - CADDS parts, storing 4-4
- EDMCASE environment variable 4-2
  - CADDS parts, storing 4-4
  - case-sensitivity in UNIX 3-5
- edmgui command (Vault GUI) 2-3
- Edmgui Error box 5-9
- Edmgui Information dialog box 2-5
- Edmgui resource file A-2
- Empty file sets, managing 5-9
- Enumeration (attribute data type) 6-5
- Environment, specifying (storing parts) 4-4
- Exiting Vault 2-5
- Explode command (binder hierarchy) 6-33
- export command C-6

## F

- File name, reserving in Vault 2-8
  - concept 4-3
  - file set, adding to 5-5
- File Set Member List dialog box 5-3
- File Sets 3-2

- checking in 4-14
  - rules 5-9
- checking out 4-9
  - rules 5-9
- creating 5-2
- deleting 5-8
- empty 5-9
- marking (delete) 5-8
- members, adding 5-3
- members, eliminating 5-6
- naming 5-2
- null 5-9
- objects in multiple 5-4
- recursive 5-3
- reserved file names, adding 5-5
- UNIX names (checked out objects) 4-12
- unmarking (delete) 5-7

File Sets browser 5-3

Files 4-2

- .attr suffix (CADDS parts) 4-6
- 00 type 4-6
- access authority 2-2
- adding to file set 5-3
- adding to stored parts 4-6
- attribute file (user defined) 3-12
- classification, defining 3-9
  - changing 3-17
- classifications 3-9
- copies, creating 2-11
- duplicating 2-11
- groups of, managing 5-2
- naming 3-6
  - conventions 3-5
- password, defining 3-10
  - changing 3-18
- password-protected, accessing 3-10
  - marking (delete or archive) 4-18
- private 3-9
- project, definition of 3-9
- properties, defining 3-6
  - changing 3-15
  - viewing 3-14
- public 3-9
- released, modifying 3-16
- removing from file set 5-6
- revision, defining 3-7
  - changing 3-16
- status level, defining 3-8
  - changing 3-17

- storage pool, selecting 3-11
  - changing 3-18
- storing locally 2-6
- transparent access 2-20
- UNIX names (checked out files) 4-11
- unmarking 4-19
- user defined attributes 3-12
- Flowchart, binder process 6-17
- Folders, design review (binders) 6-2
- Fonts, customizing in Vault A-4
  - Edmgui file, editing A-6

## G

- get command C-6
- Graphical User Interface (GUI) 2-2, 2-3
- Grouping objects (projects) 3-9
  - properties, changing 3-19
- Groups, attribute (binders) 6-5
- GUI (Graphical User Interface) 2-2
  - customizing A-1
  - exiting 2-5
  - invoking 2-3

## H

- Hierarchy (binders), viewing 6-33

## I

- ID for Vault login 2-2
- Incomplete objects, marking (delete or archive) 4-19
- Instance Attributes dialog box 6-32
- Instances (relations) 6-3
  - attributes 6-6
  - creating 6-32
  - defining 6-20
- Integer (attribute data type) 6-5

## K

- Kanji characters in password 2-2

- Keep Lock option, Check In command 2-9

## L

- Life cycle of Objects 2-16
- Local Directory, storing
  - checked out objects 4-9
- Local directory, storing 4-3
  - changing default 2-7
  - specifying (Preferences) 2-10
- Local Files browser 2-6
  - adding files 4-6
  - local directory, changing default 2-7
  - objects, naming 3-6
- Local files, storing 4-2
  - adding to stored part 4-6
  - naming 3-6
- Local name 3-6
- Local Rule 3-10
  - parts, storing 4-4
- Lock Only option, Check Out command 2-9
- Lock option, Check Out command 2-8
- Logging in to Vault 2-3
  - command line 2-4

## M

- Majority review definition 2-17
- Mandatory attribute groups (binders) 6-7
  - defining 6-20
- Marking Objects 4-17
- Members, adding to binders 6-29
- Members, adding to file set 5-3
- Members, removing from file set 5-6
- Menu commands, table of 2-8
- Menus, red text boxes in 2-3
- Messages 2-12
  - accessing 2-3
  - Notification 2-12
  - reading 2-14
  - review process 2-17
  - sending 2-12
- Mixed Notification Subscription 2-17
- Multiple file sets, including objects in 5-4
- Multiple files, managing 5-2
- Multivalued attributes 6-9

## N

- Naming objects 3-6
  - checked out 4-9
  - conventions in Vault 3-5
  - duplicates 4-16
  - file sets 5-2
- New Class Attributes dialog box 6-27
- Notification messages 2-12
- Notification Subscription definitions 2-17
- Null file sets, managing 5-9
- Number of rejects review definition 2-17

## O

- Objects 3-2
  - accessibility, restricting 4-17
  - attribute file (user defined) 3-12
  - checking in 4-13
    - menu commands, Vault 2-8
  - checking out 4-8
    - menu commands, Vault 2-8
  - classes 6-2
  - classification, defining 3-9
    - changing 3-17
  - comparison 3-4
  - copies, creating 4-15
  - deleting (Mark command) 4-17
  - design cycle 2-16
  - distributing 2-21
  - duplicating 4-15
  - grouping (projects) 3-9
    - properties, changing 3-19
  - life cycle 2-16
  - logically related (binders) 3-3
  - marking (delete or archive) 4-17
  - multiple file sets, members of 5-4
  - naming 3-6
    - conventions 3-5
    - duplicates 4-16
  - password, defining 3-10
    - changing 3-18
  - password-protected, accessing 3-10
    - marking (delete or archive) 4-18
  - properties, defining 3-6
    - changing 3-15
    - viewing 3-14

- registering 2-20
  - released, modifying 3-16
  - revision, changing 3-16
  - revision, defining 3-7
  - status level, defining 3-8
    - changing 3-17
  - storage pool, selecting 3-11
    - changing 3-18
  - storing in Vault 4-2
    - local storage 2-6
  - transparent access 2-20
  - types 6-4
  - unmarking 4-19
  - user defined attributes 3-12
- Optional Attribute groups (binders) 6-9

## P

- Packages, work (binders) 6-2
- parent child relations (binders) 6-3
- Parts 3-2
  - access authority 2-2
  - adding local files to 4-6
  - adding to file set 5-3
  - checking in 4-13
  - naming 3-10
  - other software 3-2
    - invoking from Vault (Custom menu) A-3
    - storing 4-6
  - removing from file set 5-6
  - storing 4-4
- Password-protected objects, accessing 3-10
  - marking (delete or archive) 4-18
- Passwords 3-10
  - changing 3-18
- Personalizing the GUI A-2
- Placeholder 4-8
  - reserving file names 2-8
- Pool, Storage 3-11
  - changing 3-18
- Preferences menu (customizing the GUI) A-4
- Preferences, setting 2-10
- Printer command settings, customizing in Vault A-4
- Printing documentation from Portable Document Format (PDF) file xv
- Private files 3-9
- Project ID 3-10

Projects 3-9  
 authority schemes requirement 3-9  
 files 3-9  
 Properties, defining object 3-6  
 changing 3-15  
 viewing 3-14  
 Public authorities 2-2  
 files 3-9  
 purge command C-6

## Q

Queries, changing default A-4

## R

Read access authority 2-2  
 read command C-6  
 readattr command C-6  
 readmsg command C-6  
 Real (attribute data type) 6-5  
 Recursive file sets 5-3  
 Red text boxes 2-3  
 register command C-6  
 Registering objects 2-20  
 Registration Scope 2-20  
 Rejects, number of (review definition) 2-17  
 Relations 6-3  
 aggregations 6-12  
 associations 6-11  
 attributes 6-6  
 creating 6-32  
 types of 6-11  
 Relationships 6-3  
 types of 6-11  
 Release/Revision control cycle 2-16  
 Released objects, modifying 3-16  
 remfs command C-6  
 remmas command C-6  
 remmfs command C-6  
 remmul command C-7  
 removing  
 binders 6-37  
 remup command C-7  
 remusa command C-7  
 remvlist command C-7

remvlmem command C-7  
 replace command C-7  
 reqrvw command C-7  
 Request for review, responding to 2-18  
 Reserve command 2-9  
 reserve command C-7  
 Reserving file names 2-8  
 concept 4-3  
 file sets, adding to 5-5  
 reset command C-7  
 Resource file, edmgui (customizing Vault GUI) A-2  
 Restricting accessibility for objects (Mark command) 4-17  
 Results of review, learning 2-19  
 Review 2-16  
 approval schemes 2-17  
 binders 6-14  
 definitions 2-17  
 initiating 2-19  
 list of reviewers 2-16  
 process, tracking 2-16  
 request, responding to 2-18  
 results, learning 2-19  
 setting up 2-19  
 Review folders, design (binders) 6-2  
 Revision code 2-16  
 changing 3-16  
 Revision control cycle 2-16  
 Revisions (object properties) 3-7  
 changing 3-16  
 Root binder 6-3  
 rsvp command C-7

## S

Save As dialog box (duplicating files) 2-11  
 Vault attributes, specifying 3-11  
 Scopes, selection 4-6  
 Selection scopes 4-6  
 sendmsg command C-7  
 Sequential Notification Subscription 2-17  
 Server name, Vault 2-2  
 Signing off Vault 2-5  
 Signing on to Vault 2-3  
 command line 2-4  
 User ID, changing 2-5  
 Vault, changing 2-5

- Signing out binders 6-14
- signoff command C-8
- signon command C-8
- signout command C-8
- Software, CAD/CAM 3-2
  - invoking from Vault (Custom menu) A-3
  - storing in Vault 4-6
- Status codes 2-16
- Status levels 3-8
  - changing 3-17
  - revision (control cycle) 2-16
- Storage pool 3-11
  - changing 3-18
- Store command 2-8
- store command C-8
- Store in Vault dialog box 2-6
  - attributes, specifying 3-11
  - Leave Signed Out button 4-2
  - object properties, specifying 3-6
- Stored parts, adding local files to 4-6
- Storing objects (files, parts, and directories) 4-2
  - custom parts (other software) 4-6
  - local storage 2-6
- String (attribute data type) 6-5
- Subcategories, relations (binders) 6-4
- Subscription definitions, Notification 2-17
- System administrator settings A-2

## T

- Tracking the review process 2-16
- Transactions in Vault, overview 2-6
- Transparent file access 2-20
  - distributing on demand 2-21
- Type Definition dialog box 6-19
- Type Definitions (binders)
  - defining 6-18
- Type definitions (binders) 6-4

## U

- Unanimous review definition 2-17
- UNIX names, checked out objects 4-11
- UNIX, case-sensitivity (file naming conventions) 3-5
- Unlock option 2-9
  - Check In command 2-9

- Unmark command 4-19
- unmark command C-8
- Unmark Marked Files dialog box 4-19
- Unmarking objects 4-19
  - file sets 5-7
- update command C-8
- User defined attributes 3-12
- User ID 2-2
- User settings (Preferences menu) A-2

## V

- Vault 1-2
  - accessing 2-3
  - attributes 3-11
  - changing sign on 2-5
  - character conversions 4-5
  - command line C-1
  - connecting to 2-2
  - customizing GUI A-1
  - distributed (DV) 2-20
  - exiting 2-5
  - file names, reserving 2-8
  - functions 1-3
  - GUI, invoking (edmgui command) 2-3
  - logging in 2-3
  - menu commands (table) 2-8
  - messages 2-12
  - naming conventions 3-5
  - object properties, defining 3-6
    - changing 3-15
    - viewing 3-14
  - revision, changing 3-16
  - selection scopes 4-6
  - server name 2-2
  - signing off 2-5
  - storage pool 3-11
    - changing 3-18
  - storing objects 4-2
    - locally 2-6
  - transactions, overview 2-6
  - transparency 2-20
- Vault Files browser 2-3
  - checking in files 2-11
  - checking out files 2-10
  - object properties, changing 3-16
  - reserved file name, removing 2-8
- veruattr command C-8

View, distributed (registering an object) 2-20

## W

Work packages (binders) 6-2

working with binders 6-25

Write access authority 2-2

