

# Locator/PC User Guide

---

Optegra<sup>®</sup> Release 6

DOC40025-011

---

**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

Book Conventions _____	xvi
Online User Documentation _____	xvii
Printing Documentation _____	xvii
Resources and Services _____	xviii
Documentation Comments _____	xviii

## Introduction

Overview of Locator _____	1-2
Understanding the Distributed Vault _____	1-3
Understanding Files, Parts, File Sets, and Binders _____	1-4
Classifying Files _____	1-5
Authority Schemes _____	1-5
File Attributes _____	1-5
File Operations _____	1-6
License Requirements _____	1-7

---

## Running Locator

Starting Locator _____	2-2
The Toolbar _____	2-2
The Locator Icons _____	2-3
Accessing Items in the Vault _____	2-5
Specifying a Query for the Vault Browser _____	2-6
Accessing the Vault File Set Browser _____	2-7
Specifying a SQL Pattern _____	2-8
Creating a File Set _____	2-8
Adding to a File Set _____	2-8
Accessing Items in the Distributed Vault _____	2-9
Using Units of Measure _____	2-10
Limitations and Considerations _____	2-10
Accessing Units of Measure _____	2-11
Viewing Audit Files _____	2-12
Viewing Query Results Files _____	2-13
Customizing Features for Download and Check Out _____	2-14
Destination _____	2-15
Disposition _____	2-15
Configuring Locator _____	2-16
Displaying Messages from the Vault _____	2-16
Changing Vault Browser Columns _____	2-16
Accessing Help Information _____	2-17
Exiting from Optegra Applications _____	2-18

---

## Locator Toolbar Options

Vault Login _____	3-3
Vault Logoff_____	3-4
Optegra Login _____	3-5
Optegra Logoff _____	3-6
Changing the User Password _____	3-7
Configuring Vault Hosts, Domain, and Port _____	3-8
Show Log Window _____	3-9
Saving the Log to a File _____	3-9
Clearing the Log _____	3-9
Vault Browser _____	3-10
File Set Browser _____	3-11
Specifying a SQL Pattern _____	3-11
Creating a File Set _____	3-11
Adding to a File Set _____	3-12
Store Item _____	3-13
Reserve Name _____	3-16
Copy Item _____	3-17
Check Out Item _____	3-18
Check In Item _____	3-20
Download Copy of Item _____	3-21
Update Item _____	3-22
Change Info about Item _____	3-23
Reset Item _____	3-25
Purge Item _____	3-26
Mark Item _____	3-27
Unmark Item _____	3-29
Read Messages _____	3-30

---

Send Message _____	3-33
Request Review of Item _____	3-34
Vote for/against Item _____	3-36
Review Comments _____	3-37
Binder Browser _____	3-38
Register Item _____	3-39
Unregister Item _____	3-40
Add Notification _____	3-41
Delete Notification _____	3-43

## Working with Binders

Understanding Binder Concepts _____	4-2
Relation _____	4-2
Setting Up and Using Binders _____	4-3
Adding an Item _____	4-3
Specifying Binder Class Attribute Values _____	4-4
Adding a Member to a Binder _____	4-6
Specifying Relation Attribute Values _____	4-7
Deleting a Binder or a Binder Member _____	4-8
Assigning Class Attributes _____	4-8
Changing Attributes of a Binder _____	4-9
Changing Attributes of a Relation _____	4-9
Changing Information about an Item _____	4-9
Change Binder Attributes _____	4-10
Change Description of Relation _____	4-10
Storing a New Item _____	4-10
Updating the Display _____	4-10
Selecting a User-defined Query _____	4-11
Create Query _____	4-12
Deleting a Condition _____	4-13
Integrating the Binder Browser _____	4-14

---

## Locator Programmer Reference

Configuring the Initialization File _____	5-2
Locator Directories _____	5-3
Parameters Set within Locator _____	5-4
Field Display Definitions _____	5-7

## Query Facility System

Queries in Locator _____	6-2
Standard IQFs in Locator _____	6-3
Overview of IQFs _____	6-5
Example of an IQF Query _____	6-5
IQF Keywords _____	6-6
Examples of IQF Keywords _____	6-7
Selection Query Files (SQFs) _____	6-8
Types of SQFs _____	6-8
Writing SQF Files _____	6-9
Internal Queries (RQFs and HQFs) _____	6-10

## Application Programmer Interface

Overview of the Application Programmer Interface (API) _____	7-2
Differences Between Vault Programming and Locator API _____	7-3
Limitations for Command Keyword Options _____	7-3
Locator API Parameters _____	7-3
Locator API Sessions _____	7-3
API Calling Interfaces _____	7-4
IQF Support _____	7-5
Locator API Logging Facility _____	7-6
Custom Locator API _____	7-7

---

Description of Locator API Calls _____	7-8
EDMaddfs( ) _____	7-8
Syntax _____	7-8
Parameters Description _____	7-8
EDMaddmfs( ) _____	7-9
Syntax _____	7-9
Parameters Description _____	7-9
EDMchgfa( ) _____	7-9
Syntax _____	7-9
Parameters Description _____	7-10
EDMchgfcl( ) _____	7-10
Syntax _____	7-10
Parameters Description _____	7-11
EDMchgfpw( ) _____	7-11
Syntax _____	7-11
Parameters Description _____	7-12
EDMchgfrev( ) _____	7-12
Syntax _____	7-12
Parameters Description _____	7-13
EDMchgfsc( ) _____	7-13
Syntax _____	7-13
Parameters Description _____	7-14
EDMchgupw( ) _____	7-14
Syntax _____	7-14
Parameters Description _____	7-15
EDMciedm( ) _____	7-15
Syntax _____	7-15
Parameters Description _____	7-15
Example _____	7-16
EDMcopy( ) _____	7-16
Syntax _____	7-16
Parameters Description _____	7-17
EDMcpdm( ) _____	7-17
Syntax _____	7-17
Parameters Description _____	7-18
EDMend( ) _____	7-18
Syntax _____	7-18

---

EDMfedm( ) _____	7-18
Syntax _____	7-18
Parameters Description _____	7-19
EDMgetatlst( ) _____	7-19
Syntax _____	7-19
Parameters Description _____	7-19
EDMgetopt( ) _____	7-20
Syntax _____	7-20
Parameters Description _____	7-20
EDMinit( ) _____	7-20
Syntax _____	7-20
EDMiqf( ) _____	7-20
Syntax _____	7-20
Parameters Description _____	7-21
EDMlocalname( ) _____	7-21
Syntax _____	7-21
Parameters Description _____	7-22
EDMlog( ) _____	7-22
Syntax _____	7-22
Parameters Description _____	7-22
EDMmarka( ) _____	7-23
Syntax _____	7-23
Parameters Description _____	7-23
EDMmarkd( ) _____	7-23
Syntax _____	7-23
Parameters Description _____	7-24
EDMmarkr( ) _____	7-24
Syntax _____	7-24
Parameters Description _____	7-25
EDMmenu( ) _____	7-25
Syntax _____	7-25
Parameters Description _____	7-26
EDMpurge( ) _____	7-26
Syntax _____	7-26
Parameters Description _____	7-27
EDMrcustom( ) _____	7-27
Syntax _____	7-27
Parameters Description _____	7-27

---

EDMread( ) _____	7-28
Syntax _____	7-28
Parameters Description _____	7-28
EDMreadmsg( ) _____	7-30
Syntax _____	7-30
Parameters Description _____	7-30
EDMremfs( ) _____	7-30
Syntax _____	7-30
Parameters Description _____	7-31
EDMremmfs( ) _____	7-31
Syntax _____	7-31
Parameters Description _____	7-31
EDMreplace_dv( ) _____	7-31
Syntax _____	7-31
Parameters Description _____	7-32
EDMreqrvw( ) _____	7-32
Syntax _____	7-32
Parameters Description _____	7-33
EDMreserve( ) _____	7-33
Syntax _____	7-33
Parameters Description _____	7-34
EDMreset( ) _____	7-34
Syntax _____	7-34
Parameters Description _____	7-35
EDMrsvp( ) _____	7-35
Syntax _____	7-35
Parameters Description _____	7-35
EDMsendmsg( ) _____	7-36
Syntax _____	7-36
Parameters Description _____	7-36
EDMsetopt( ) _____	7-36
Syntax _____	7-36
Parameters Description _____	7-37
EDMsignoff( ) _____	7-37
Syntax _____	7-37
EDMsignon( ) _____	7-37
Syntax _____	7-37
Parameters Description _____	7-38

---

EDMsignout( ) _____	7-38
Syntax _____	7-38
Parameters Description _____	7-39
EDMstore( ) _____	7-39
Syntax _____	7-39
Parameters Description _____	7-40
EDMsvedm( ) _____	7-41
Syntax _____	7-41
Parameters Description _____	7-41
EDMunmark( ) _____	7-41
Syntax _____	7-41
Parameters Description _____	7-42
EDMupdate( ) _____	7-42
Syntax _____	7-42
Parameters Description _____	7-42
Descriptions of Custom API Calls _____	7-43
EDMccpdm( ) _____	7-43
Parameters Description _____	7-43
EDMccend( ) _____	7-43
Parameters Description _____	7-44
EDMccfedm( ) _____	7-44
Parameters Description _____	7-44
EDMccinit( ) _____	7-44
Parameters Description _____	7-45
EDMcclocalname( ) _____	7-45
Syntax _____	7-45
Parameters Description _____	7-45
EDMccmenu( ) _____	7-45
Parameters Description _____	7-46
EDMccsetopt( ) _____	7-46
Parameters Description _____	7-46

## Working with Other Applications

Rulebases _____	8-2
DDE Support _____	8-4

---

# Administering Vault Desktop Server

Overview of the Vault	
Desktop Server Configuration File _____	9-2
Desktop Support and Configuration _____	9-2
Server Locations _____	9-2
Log File Locations and Names _____	9-2
Configuration Basics for the Desktop Server _____	9-3
Categories of Configuration Parameters _____	9-4
Protocol Parameters _____	9-5
TCP-port port number _____	9-5
TCP-keep-alives ON OFF _____	9-5
Temp-directory temporary directory path _____	9-5
Timer, Interval, and Limit Parameters _____	9-6
Activity-timeout n _____	9-6
Inactivity-timeout n _____	9-6
Configuration-check-interval n _____	9-7
Log-file-reset n _____	9-7
Log-file-size n _____	9-7
Log-file-retention n _____	9-7
Tracing Parameters _____	9-8
Verbose ON OFF _____	9-8
Trace-commands ON OFF _____	9-8
Trace-recv-data ON OFF _____	9-8
Trace-send-data ON OFF _____	9-9
Trace-signed-out ON OFF _____	9-9
Trace-custom ON OFF _____	9-9

## Messages and Troubleshooting

Overview of Messages _____	A-2
Message Descriptions _____	A-3

## Glossary

---

# Preface

---

The *Locator/PC User Guide* provides information to help you access one or more Vaults using a client-server system. Locator runs on Windows NT.

Once signed on to the Vault server, Locator allows you to do the following:

- Browse, use, and query files on a Vault or Distributed Vault
- Participate in reviews
- Send, read, and respond to electronic messages
- Organize your parts, files, and objects into file sets and binders
- Launch a local application, such as a spreadsheet program, directly after reading or checking out an item

You can customize the following interface elements of Locator:

- Menus
- Views
- Application operations

Finally, the system administrator can also customize Locator commands. The Locator/PC Programming SDK, available separately, can be needed for advanced customization features.

## 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*.

# Introduction

---

This chapter provides an overview of Locator/PC and discusses the following topics:

- Overview of Locator
- Understanding the Distributed Vault
- Understanding Files, Parts, File Sets, and Binders
- Classifying Files
- License Requirements

# Overview of Locator

Locator, working with Vault, combines relational database management technology (RDBMS) with task-oriented applications to control, manage, and distribute electronic documents. These documents can be created by any computer application including computer-aided engineering (for example, CAD, and CAM), word processing, and spreadsheets.

Optegra applications are a client-server system. After you have signed on to the Vault server using an account and password, you can do the following with Locator:

- Write, read, and delete files, parts, and file sets from local or remote vaults
- Display and change system information about a file, part, or file set, or binder (metadata)
- Create binders, a collection of logically related objects, for quick access and review
- Read and send Vault mail messages
- Query the Vault using a query (IQF) file
- Request a review of a file, part or file set
- Vote in a review

## Understanding the Distributed Vault

With the Info Queries option in the Vault browser, you can enter the name of the object for the query Locate Object on the Distributed Vault. The system displays any occurrence of the object on the various vaults in the Distributed Vault environment, as well as Vault ID information.

The Distributed Vault gives you transparent access to objects on remote Vaults without your having to know the location of the object. The Distributed Object Directory (DOD) provides access to objects that you register for use on the Distributed Vault. The Distributed Vault supports the following commands: Register, Subscribe and Locate.

Please note: The Subscribe command is referred to as notify (Add Notification, Delete Notification) throughout this document.

The Register Item and Unregister Item options use the Register command. The Add Notification and Delete Notification use the Notify command.

# Understanding Files, Parts, File Sets, and Binders

The file is the fundamental unit of Locator. You can also use files organized into larger units, for example, parts, file sets, and binders. You can access parts, file sets and binders with Locator. All file commands work with parts. Definitions of parts, file sets, and binders follow. You define all groups of files within the Vault.

Please note: You use the Vault File Set Browser to create a file set, and the Vault Binder Browser to create a binder.

A part is a logical group of files, defined within an application, such as CAD/CAM, and treated as a unit. A part cannot contain another part. A part has its own Locator attributes that each file in the part shares. A file can belong to only one part. Custom part definitions can also be used in Vault.

A file set is a group of files treated as a single unit within Locator. For example, you can create a file set of a CADD part and all related text files. A file set can contain files, parts, and other file sets. The set members can belong to more than one classification (public, private or project) or more than one file set. Using file sets ensures that all member files contain pertinent information. This can save your time when you want to perform the same operation on related files. The system performs operations on the members depending upon each file's authority scheme.

A binder is a container for building a collection of logically related objects. Binders can contain files, parts or other binders. Defining and creating relationships among different objects in the Vault allow you quick access and review of your related collection of data. Whereas file sets treat all files as a single unit, binders treat each object according to prescribed relationships and attributes. You set up the relationships as you add each member to the binder.

## Classifying Files

You can store a file in the Vault as one of the following three classifications:

- Public
- Private
- Project

The Project classification is further subdivided into any number of projects. For a given user, the Vault can be represented like this:

Project 1	Project 2	Project 3	Project n
Project 1 files	Project 2 files	Project 3 files	Project n files

## Authority Schemes

Files are governed by two types of authority schemes: public and project. Each authority scheme has its own users, read and write accesses, and command lists. System and project administrators are responsible for setting up and maintaining authority schemes. Each project has its own authority scheme.

Private files do not have a separate authority scheme, but rather follow the public authority scheme.

Public Authority Scheme		Project Authority Schemes			
		1	2	3	...n
Private files	Public files	Project 1 files	Project 2 files	Project 3 files	Project n files

## File Attributes

Locator does more than simply store a file. Locator maintains attributes, or metadata, about the file, such as its name, owner, date of last change, project, if any, and so on. Binders have their own class attributes explained in Chapter 4, “Working with Binders” Your system administrator can also define custom attributes for files.

## File Operations

The file is the basic entity in Vault. A user can perform several different operations on a file.

The first step is to store the file in the Vault. After that, the file can be read, deleted, or modified. Locator provides several different ways to read, write or delete a file.

When you check out a file, the system puts a lock on any simultaneous modification of that file. Another user can request a copy to read but cannot check the file out until it has been checked back in. Only the user who checked out the file, or a user with administrative privileges, can check it back in.

The following table shows all file commands and some of their effects. Some commands affect the file's metadata.

**Table 1-1** Locator Commands and Their Effects

Locator Commands	Vault File Operations	File Metadata Changes
STORE	read local file, write to Vault	description
COPY	read Vault file, new write to Vault	description
RESERVE		description, optional check-out
Mark (MARKR, MARKA, MARKD)	read Vault tape, write to Vault	
Download Copy (READ)	read Vault file, write local copy	
Check Out (GET)	read Vault file, write local copy	check out
RESET		unlock
UPDATE	read local file, write to Vault	
Check in (REPLACE)	read local file, write to Vault	unlock
Change Info (various attributes)		description
PURGE	delete (now)	
Request Review (REQRVW)		lock

When you read or check out a file, Locator makes a copy of the file and places it on your node. After working with the local copy, if you want to save your changes to the original file in the Vault, you choose the Update Item or Check In Item option.

## License Requirements

Locator/PC is licensed software that requires an “epd” licensing daemon. This grants you permission to use the application. For information on setting a license daemon on your license host, refer to *Using the License Manager*.

Ensure that a copy of the license file exists in `c:\cvswlm\epd\epd.lic` or set the `LM_LICENSE_FILE` environment variable. For more information, refer to *Using the License Manager*.



# Running Locator

---

This chapter contains basic information about running Locator/PC.

- Starting Locator
- Accessing Items in the Vault
- Using Units of Measure
- Viewing Audit Files
- Viewing Query Results Files
- Customizing Features for Download and Check Out
- Configuring Locator
- Accessing Help Information
- Exiting from Optegra Applications

## Starting Locator

To start Locator directly, complete the following steps:

1. Choose Start >Programs> EPD.Connect and Optegra > Locator. The Optegra Locator toolbar appears as shown in the next section.
2. Choose any option to get the Optegra Log in window.

Please note: If you want to use more than one Optegra application, choose Optegra Login from the toolbar.

### The Toolbar

The Optegra Locator toolbar includes 31 icons for the Locator commands for access to items stored in a Vault. The Optegra Locator toolbar is shown in next graphic. The distributed vault icons (the last four) are available only if you have signed on to a vault with a distributed environment.



Once you have signed on to Locator or used the Optegra Login option, select any command from the toolbar. A window appears when you select any option except Vault Logoff and Optegra Logoff.

## The Locator Icons

Optegra Locator toolbar provides icons for operations within Locator and for the Vault. The command options are listed in the following table:

**Table 2-1 Locator Command Options**

<b>Command Option</b>	<b>Description</b>
Vault Login	Sign on to the Vault
Vault Logoff	Sign off of from the Vault
Optegra Login	Connect to Optegra applications using the common log in
Optegra Logoff	Disconnect from all Optegra applications
Change User Password	Create a new user password
Vault Host and Domain Configuration	Store names of multiple hosts and domains
Show Log Window	Display the Local Optegra Locator transaction log
Vault Browser	Access the Vault items (file or part) in the Vault
File Set Browser	Access the Vault file sets in the Vault
Store Item	Store a new file, or part in the Vault
Reserve Name	Reserve a name for the future storing of a file
Copy Item	Create a copy of a Vault item in the Vault
Check Out Item	Obtain a local copy for modification, locking the original object in the Vault to prevent modification by others
Check In Item	Return an object to the Vault, saving changes to the original, and releasing it for others to modify
Download Copy of Item	Obtain a local copy for read only purposes, leaving the original object available for others to check out
Update Item	Save changes to the original object in the Vault while keeping the object checked out
Change Info about Item	Change the attributes contained in the metadata of an object within the Vault
Reset Item	Return a checked out object to the Vault without saving any changes. (Cancels a check out operation)
Purge Item	Delete a file or part immediately
Mark Item	Mark a file, part or file set for archiving, deleting, or restoration from tape
Unmark Item	Remove a mark from a file, part or file set
Read Messages	Retrieve and display Vault messages
Send Message	Send a message to an individual user or list of users
Request Review of Item	Request that an object be placed in the review process or advanced to the next status level
Vote for/against Item	Approve or disapprove items currently in the review process.

**Table 2-1 Locator Command Options**

<b>Command Option</b>	<b>Description</b>
Review Comments	Display the review comments about an object
Binder Browser	Access the binders in the Vault
*Register Item	Register an Item to DOD
*Unregister Item	Unregister an Item from DOD
*Add Notification	Establish a relationship between an event and an action
*Delete Notification	Delete a Notification

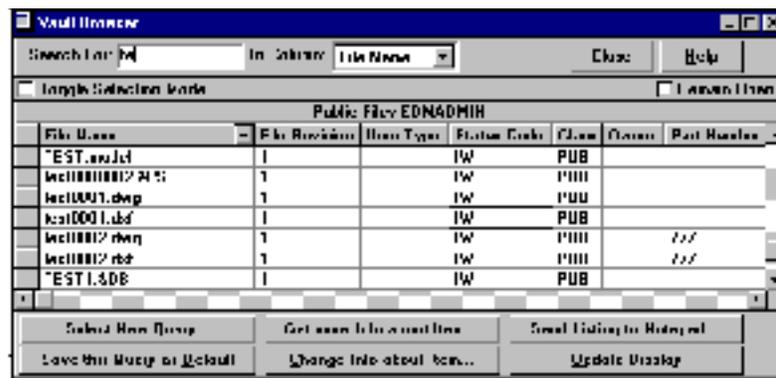
\*The last four icons are functional only if you are signed on to a distributed vault.

## Accessing Items in the Vault

Locator supports different views of data using your choice of a file browser to access items in the Vault or the Distributed Vault. For information using the Distributed Vault see “Accessing Items in the Distributed Vault” on page 2-9. You can use the following browsers:

- Vault Browser for files and parts (See “Specifying a Query for the Vault Browser” on page 2-6.)
- Vault File Set Browser for file sets (See “Accessing the Vault File Set Browser” on page 2-7 and “File Set Browser” on page 3-11.)
- Binder Browser for binders (See “Binder Browser” on page 3-38.)
- Custom Browser for items and views with your own parameters

To configure Locator to use one of these browsers as the default, on the Optegra Locator Log select Browser > Vault Browser or any browser of your choice from the list. The specified browser appears as part of the execution of any future operation on items stored in the Vault.

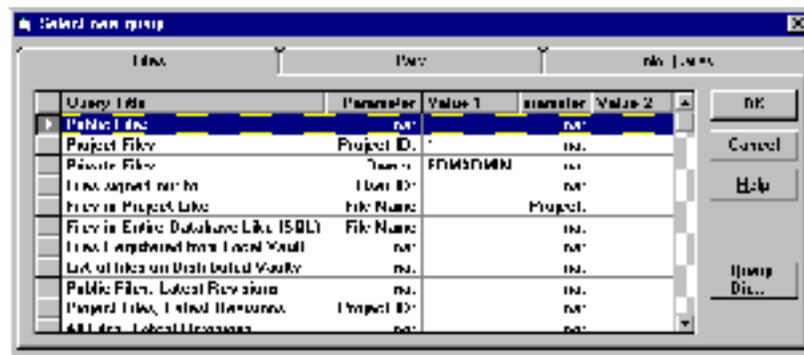


Many commands activate the Vault Browser. From the Vault Browser you can select the files or parts on which to perform operations. You can also use the Search For text box to enter characters in the object name. Any matching items can be identified by Locator. (For information on the other browsers, see previous bulleted list.)

## Specifying a Query for the Vault Browser

Each view or list of items in the Vault Browser results from a query. To see a different list, perform the following steps:

1. Click Select New Query from the bottom of the Vault Browser. The Select New Query window appears.



2. Specify the type of object to be listed by selecting either the Files, Parts, or Info Queries tab from the top of the panel.

Eleven default listings apply to files or parts:

- Public Files lists all files or parts that are classified as belonging to the Public Project in the Vault.
- Project Files lists all files or parts that are classified within a specified Project within the Vault.
- Private Files lists all files or parts that are classified as Private files for the current Locator user ID.
- Files Signed Out To lists all files or parts that are currently signed out from any classification from the Vault to the current Locator user ID.
- Files in Project Like lists all files or parts in the Vault that are assigned to a specified project and match the Standard Query Language (SQL) pattern. This selection asks for parameters in a secondary dialog.
- Files in Entire Database Like lists all files or parts in the Vault that match the SQL pattern.
- Files Registered from Local Vault lists all files or parts assigned to the Local Vault.
- List of Files on Distributed Vaults lists all files or parts available through the Distributed Vault directory.
- Public Files, Latest Revision lists all the public files and parts with the latest revisions only.

- Project Files, Latest Revisions lists all the files and parts from the given project with the latest revisions.
  - All Files, Latest Revision lists all the files and parts with their latest revisions.
3. Choose the query that you want. To query an object from a specific directory, see the Query Dir option that follows immediately.
  4. Enter your query criterion in the Value fields for each parameter.
  5. Click OK. The results of the query appear in the Browser.
  6. If you want to save this view as the default view in future sessions, choose Save This Query as Default from the Browser.

You can modify the column display. See “Configuring Locator” on page 2-16.

With the Query Dir option you can select a query from a directory of your choice. On the Select New Query window do the following:

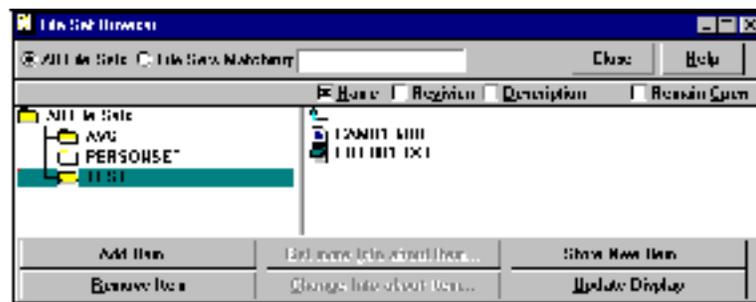
1. Choose Query Dir. The Select the Query Directory window appears.
2. Go to the required directory from your PC.
3. Click OK. You are returned to the Select New Query window.

To see a different view or list, return to step 3 under “Specifying a Query for the Vault Browser” on page 2-6.

## Accessing the Vault File Set Browser

The file set object type has its own browser panel.

1. Click the File Set Browser on the Optegra Locator toolbar. The File Set Browser appears.



2. Specify the scope of your query; you have the following choices:
  - All File Sets lists all file sets in the database.
  - File Sets Matching lists all file sets in the Vault that match the SQL pattern. SQL uses the percent (%) character for a wild card character.

## Specifying a SQL Pattern

1. Enter the SQL pattern in the File Sets Matching field.
2. Select the All File Sets field.
3. Select the File Sets Matching field again.

The system processes the query and displays the resulting file sets.

## Creating a File Set

1. On the Vault File Set Browser dialog box, with no file set selected, click Add Item. The Create File Set dialog box appears.
2. Enter the name and an optional description for the new file set and click OK.

## Adding to a File Set

1. On the Vault File Set Browser, click the file set to which you want to add.
2. Click Add Item.

The system displays a Vault browser window with the heading Select an Item to Add to File Set name.

3. Select the file(s) or part(s) you want to add. You can select the Toggle Selection Mode to enable multiple selection.
4. When you are finished selecting the items to add, click OK.

The system displays a dialog box for each item selected in succession. You can add a description in the Description field.

## Accessing Items in the Distributed Vault

You can perform Locator operations such as check out, download, and update on the items in the Distributed Vault. Procedures using the toolbar and the log window follow. The following example is for updating objects.

1. Click Update Item on the Optegra Locator toolbar.

The Select Items to Update window appears with a list of all the files and parts in the local Vault.

2. Click Select New Query. The Select New Query dialog box appears.
3. Select the Files folder to update.
4. Choose the List of files on the Distributed Vault query from the Files folder.
5. Click OK.

A list of all the files on the Distributed Vault appears in the Select Items to Reset window. The listing contains the following information:

- Vault ID: Specifies the name of the Vault on which the object resides.
  - Object Name: Specifies the name of the object.
  - Revision: Specifies the sequence number for the changed object according to the authority scheme.
6. Select one or more items that you want to update from this list. Proceed like any other task in Locator.

You can do identical operation from the Locator Log. The following example is also for updating objects.

1. On the Optegra Locator Log, select Objects > Distributed Vault.

The Vault Browser appears with lists of objects from the Distributed Vault.

2. Choose Update Item and proceed as with any other Locator option.

Please note: When you check the option Distributed Vault, the Vault Browser lists objects from the Distributed Object Directory.

## Using Units of Measure

The Unit of Measure functionality allows you to associate an unit of measurement with a numeric attribute. This can be done while:

- Assigning values to attributes
- Reading attribute values
- Changing attribute values
- Validating attribute values

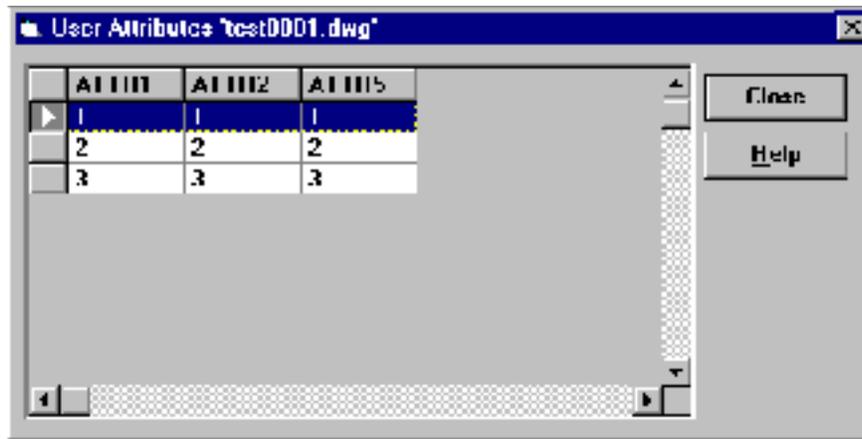
Please note: User authority is verified for assigning or changing attribute values and also reading attribute values

### Limitations and Considerations

- Units of measure can be associated with numeric attributes only.
- A numeric attribute can be optionally associated with an unit of measure.
- You have to supply the logic to convert attribute values expressed in one unit of measure to other units of measure within an unit of measure set.
- There is a default storage unit of measure associated with every attribute. While you can specify any unit of measure (belonging to the same unit of measure set as that of default unit of measure), the values are stored in terms of the default unit of measure specified in the attribute definition.
- You can read, write, or update only those user attribute values, for which you are authorized.

## Accessing Units of Measure

The User Attributes dialog box can be accessed from the Vault Browser, Store Item, Check In Item, and Update Item options of the Locator toolbar.



**Reading Attribute Values:** From the Vault Browser choose Get More Info about Item > Attributes. The User Attributes dialog box appears. You can only read the values and cannot change, when accessing the dialog box from the Vault Browser.

**Changing Attribute Values:** You can change attribute values using the Store Item, Check In Item and the Update Item options. Your authority for changing attributes is checked. For information on how to do this, see Chapter 3, “Locator Toolbar Options.”

## Viewing Audit Files

The system keeps audit information on several Locator commands and stores the results in audit files in ASCII text format. You can customize the Audit File application. Audit files are stored in the %EPD\_HOME%\audit subdirectory on the Locator drive.

The following file commands return results stored in audit files:

- Store Item
- Check Out Item
- Check In Item
- Download Copy of Item
- Update Item
- Reset Item
- Request Review
- Read Messages

Audit files are named after the Locator option and are given AUD as the file extension under Windows, for example, READ.AUD. Locator commands display a dialog box asking if you want to see the audit file. Selecting yes displays the file in the Windows Notepad text editor.

The Notepad window title includes the name of the file. These windows have their own menus. Use File from the Locator Log for operations such as printing, saving, and exiting (closing).

## Viewing Query Results Files

The Locator Query (IQF) operation also returns a result. Each IQF can have its own file for its results (the default is `RunIQF.OUT`). Query results appear immediately within the Vault File Browser. If you choose Send Listing to Notepad, you can view the query results in the ASCII text editor, Notepad. The Query Facility System is described in Chapter 6, “Query Facility System.”

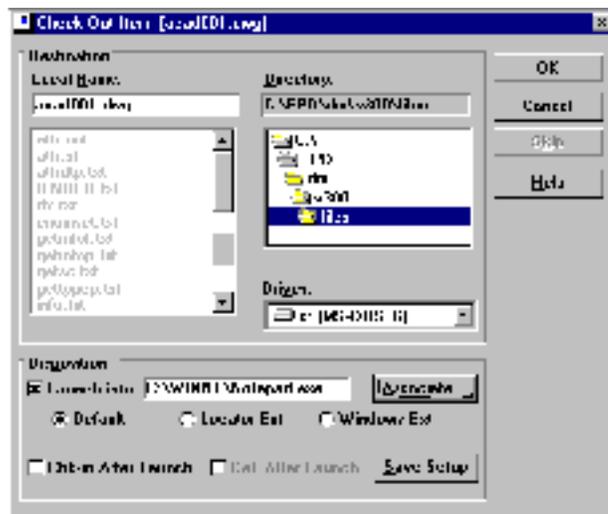
## Customizing Features for Download and Check Out

You can customize the Download and Check Out operations by specifying local file parameters. You can also specify the starting of a Windows or MS-DOS application upon making a local copy.

1. Click Show Log Window on the Optegra Locator toolbar.
2. Choose Options > Check-Out Options and one option of your choice from the following list:
  - Verify Disposition : Starts an application when an item is either checked out or a read only copy is downloaded.
  - Verify Destination : Specifies the destination directory for the local copy.
  - Verify Overwrite : Specifies whether the local copy of an item is to be deleted after a start or check out operation.
  - Shorten File Names : Automatically shortens the names of the files if they are too long for MS-DOS.

When you select any of the previous options, additional information appears on the Download and Check Out windows. You can also configure these commands by editing the `EDMWIN.INI` initialization file discussed later in this guide.

When you choose the Check-Out option and select the items to check out from the Vault Browser, the system displays the Select Items to Check Out dialog box.



## Destination

**Local Name:** Specifies the name of the file selected in the main window listing after choosing Download Copy of Item or Check Out Item.

**Directory:** Specifies the directory where the file read or checked out from the Vault is to be stored on your PC.

**Drive:** Scans all disk drives available in the local environment.

## Disposition

The check boxes in the Disposition section provide parameters for local file operations. These task parameters affect local file operations on the selected Vault file(s). The meanings of the check boxes are as follows:

- **Launch Into:** Starts the specified application upon successful check out or download of a Vault object.

The path of the application displayed in the text box depends on the option selected from the following:

- **Default :** Specifies the application against the “default” key in [extensions] section of EDMWIN.INI file.
- **Locator Ext :** Specifies the application against the file’s extension in the [extensions] section of EDMWIN.INI file.
- **Windows Ext :** Associates the application with the file’s extension in Windows environment.
- **Check-in after Launch :** Returns the object to the Vault upon completion of the started application.
- **Delete after Launch :** Removes a file from the local working space after return to the Vault.
- **Associate :** Associates an item to an application based on the file extension of the item. For example, a .TXT file might go to the Notepad application, or a .DOC file to Word for Windows.
- **Save Setup :** Records settings in the EDMWIN.INI initialization file to be used as the future defaults.

# Configuring Locator

You can configure the display of messages from the Vault and change the Vault Browser columns.

## Displaying Messages from the Vault

The Vault sends messages after every command, whether the command is successful or unsuccessful. Unsuccessful messages are always shown to the user in a message box. By default, success messages are not shown.

You can configure the default setting for messaging in one of these ways:

- Modify the `EDMWIN.INI` initialization file
- Select Optegra Log > Options > Confirmation Messages

## Changing Vault Browser Columns

You can change the names and the order of the columns in the Vault Browser. Each type of listing (files, parts, binders and file sets) has its own set of columns. You can add and remove columns for a listing only by changing the SQF files, as discussed in “Selection Query Files (SQFs)” on page 6-8.

## Accessing Help Information

Locator features different kinds of help information. To familiarize yourself with the toolbar options, select Control-menu box > Tool Tips and names of the options appear on the status bar. You can also use the Help buttons on each window. From the Optegra Locator Log Help menu item you can choose the Locator > Browser User Guide to get the latest HTML user guide.

Please note: To use the *Locator/PC User Guide*, you must have both a HTML browser and the user guide files installed on your system.

The next chapter describes all the Locator options in their order on the toolbar. The topics correspond with the Tool Tips on toolbar status line available by selecting Control-menu box > Tool Tips.

## Exiting from Optegra Applications

From the Optegra Locator toolbar, you can exit from the Vault or from all Optegra applications.

- 1. Choose Vault Logoff or choose Optegra Logoff.**

Optegra Logoff disconnects you from all Optegra applications accessed through the common log in, including the Vault.

- 2. To close the Locator toolbar, select Control-menu box > Close.**

# Locator Toolbar Options

---

This chapter provides procedures for using all the Locator options. Options are presented in their order of appearance on the toolbar. Information and tips on using the options appear at the end of many procedures. To display the Tool Tips (option names) on toolbar status bar, select Control-menu box > Tool Tips.

- Vault Login
- Vault Logoff
- Optegra Login
- Optegra Logoff
- Changing the User Password
- Configuring Vault Hosts, Domain, and Port
- Show Log Window
- Vault Browser
- File Set Browser
- Store Item
- Reserve Name
- Copy Item
- Check Out Item
- Check In Item
- Download Copy of Item
- Update Item
- Change Info about Item
- Reset Item
- Purge Item

- Mark Item
- Unmark Item
- Read Messages
- Send Message
- Request Review of Item
- Vote for/against Item
- Review Comments
- Binder Browser
- Register Item
- Unregister Item
- Add Notification
- Delete Notification

## Vault Login



You must log on to Vault to perform Locator operations such as working with Vault files. Before you sign on to the Vault for the first time, get your official user name and password from your Vault System Administrator. Then perform the following steps:

1. Click Vault Login on the Locator toolbar. The Optegra Login window is displayed.
2. Enter the user name and password.

If you have any messages, the system asks if you want to read them. For details see “Vote for/against Item” on page 3-36.

For more information on common login, refer to *Installing Oracle for Optegra Applications*.

## Vault Logoff



The Vault Logoff option closes your connection to a Vault. To log off, follow this procedure:

1. Click Vault Logoff on the Optegra Locator toolbar.

When signed off, you can only perform Locator operations such as signing on and changing your password or your Vault connection.

Please note: Before signing off, you can check if you have items signed out from the Vault. Use the Query or the standard query `SIGNOUT.IQF`. For information see Chapter 6, “Query Facility System.”

# Optegra Login



When you connect to an Optegra application, the common login uses your user name and password to connect to other Optegra applications. To log on to Optegra, perform the following steps:

1. Click Optegra Login on the Optegra toolbar. The Optegra Login window appears.
2. Enter the user name and password.
3. Choose any item from the Optegra toolbar or start the Optegra application of your choice. Your name and password is entered automatically.

If you have any messages, the system asks if you want to read them. For details see “Vote for/against Item” on page 3-36.

For more information on common login, refer to *Installing Oracle for Optegra Applications*.

## Optegra Logoff



The Optegra Logoff option closes your connection to all Optegra applications. To sign off, follow this procedure:

1. Click Optegra Logoff on the Optegra Locator toolbar. The Optegra Logout message box appears.
2. Click OK. The Optegra Locator toolbar remains in view.

When disconnected from all Optegra applications, you can only perform Locator operations such as signing on and changing your password or Vault connection.

## Changing the User Password



Locator allows you to change your assigned Vault password after you have signed on.

1. Click Change User Password on the Optegra Locator toolbar. The Change Password window appears.

A screenshot of the 'Change Password' dialog box. The dialog has a title bar with the text 'Change Password' and a close button. It contains three text input fields: 'User Name' with the value 'edmadmin', 'Current Password' with asterisks, and 'New Password' with asterisks. To the right of the input fields are three buttons: 'Change', 'Cancel', and 'Help'. At the bottom right, there is a checkbox labeled 'Show Passwords' which is currently unchecked.

2. Enter your current password and the new password.
3. Click Change.

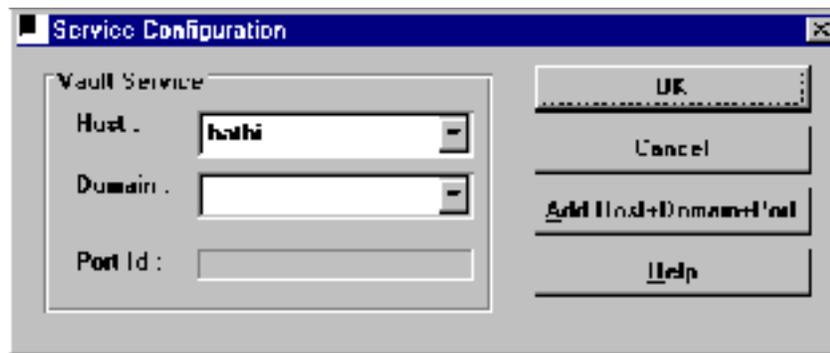
Please note: For security reasons, the password characters appear as asterisks (\*). However, you can see the actual password input by selecting the Show Passwords check box.

## Configuring Vault Hosts, Domain, and Port

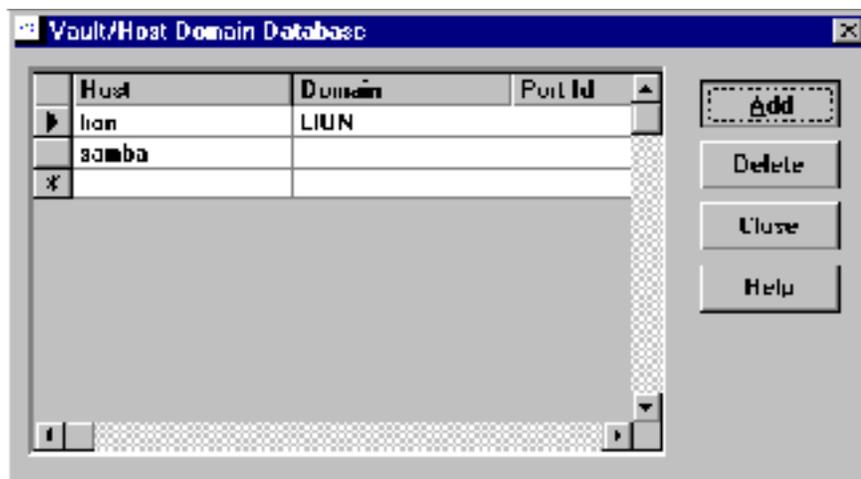


If you work with multiple hosts or domains with a specific port, you can store their names to speed up your sign-on process. To add or delete Vault hosts and domains with port combination for your configuration, perform the following steps:

1. On the Locator toolbar, click Vault Host and Domain Configuration. The Service Configuration window appears.



2. Click Add Host + Domain + Port. The Vault Host/Domain Database window appears.



- To add a name, type the new name in the Host or Domain and Port fields. Click Add.
  - To delete a host or domain, select the name to be deleted from the list and press the Delete key.
3. Click Close to return to the Service Configuration window.

Refer to “Protocol Parameters” on page 9-5 for information on ports.

## Show Log Window



By default, the Optegra Locator Log window appears when you sign on. If the log is not in view, choose Show Log Window from the toolbar. The session log appears in a separate window.



You can change the size and the location of the log window and save these changes as defaults for future Locator sessions. To close the Log window, choose File > Hide from the Optegra Locator Log window.

## Saving the Log to a File

You can save the current session log to a file at any time by performing the following steps:

1. On the Optegra Locator Log toolbar choose File > Save Log As. The Save Log as window appears.
2. Enter a new file name or use the `edmlog.txt` file.

Unless you use a new file, the system overwrites the existing file with the contents of the current session log.

## Clearing the Log

To clear the Optegra Locator Log and begin a new session log, choose File > Clear Log.

Information:

Locator records your work session, including commands and messages, in the Optegra Locator Log. The log entries help you recall the operations you have performed and their sequence. You can also use the log to reconstruct actions during debugging or training.

# Vault Browser



To select a query for the Vault, perform the following steps:

1. Click Vault Browser on the Optegra Locator toolbar. The Vault Browser appears. For more information on the Vault Browser, see “Accessing Items in the Vault” on page 2-5.
2. Choose Select New Query. The Select New Query dialog box appears.



3. Select a tab for Files, Parts, or Info Queries.
4. Select the required query.
5. Select the appropriate fields for Parameter 1 or Parameter 2. Include any values in the Value 1 and Value 2 fields.

Please note: Choose Query Dir to specify the directory for the query.

To run a query, click OK. The query results are displayed in the Vault Browser.

6. To edit the query file report, click Send Query to Notepad.

The Notepad is the default editing application for query file results. It has four menus: File, Edit, Search, and Help.

You can custom configure another editing application. You can also customize a query to access a specific viewer. See “Custom Locator API” on page 7-7.

Information:

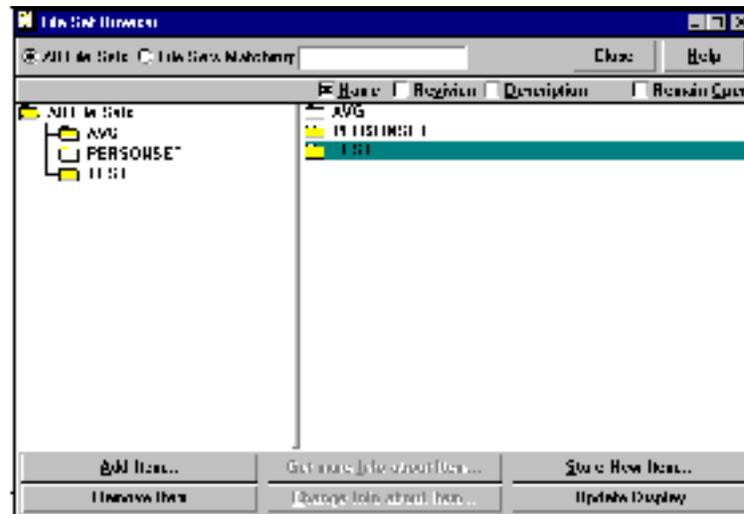
Locator provides an Interactive Query Facility (IQF) from within the Vault Browser to query the Vault for information. Queries are in the form of IQF text files. Many are provided with Locator. You can also create custom IQFs in Locator.

## File Set Browser



To use the File Set Browser, follow these steps:

1. Click the File Set Browser on the Optegra Locator toolbar. The File Set Browser appears.



2. Specify the scope of your query; you have the following choices:
  - All File Sets lists all file sets in the database.
  - File Sets Matching lists all file sets in the Vault that match the SQL pattern. SQL uses the percent (%) character for a wild card character.

### Specifying a SQL Pattern

1. Type the SQL pattern in the File Sets Matching text box.
2. Select the All File Sets option.
3. Select the File Sets Matching option. The system processes the query and displays the resulting file sets.

### Creating a File Set

1. On the Vault File Set Browser dialog box, with no file set selected, click Add item. The Create File Set dialog box appears.
2. Enter the name and an optional description for the new file set and click OK.

## Adding to a File Set

- 1.** On the Vault File Set Browser, click the file set to which you want to add an item.

- 2.** Click Add Item.

The system displays a Vault browser window with the heading **Select an Item to Add to File Set name**.

- 3.** Select the file(s) or part(s) you want to add. You can select Toggle Selection Mode to enable multiple selection.

- 4.** When you are finished selecting the items to add, click OK.

The system displays a dialog box for each item selected in succession. You can add a description in the Description text box.

## Store Item



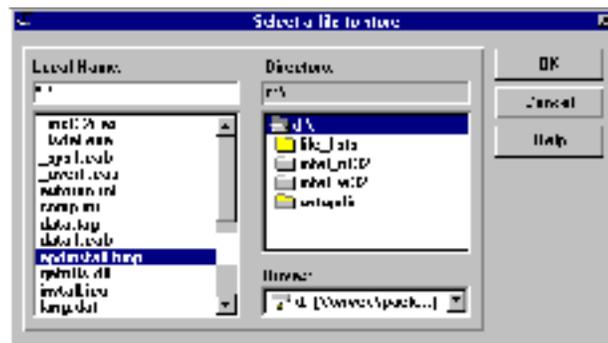
To add a file or part from your PC to the Vault, use the Store Item option as follows.

Please note: You can use Store Item only once on a file or part.

1. Click Store Item on the Optegra Locator toolbar. The Select Item Type to Store dialog box appears.



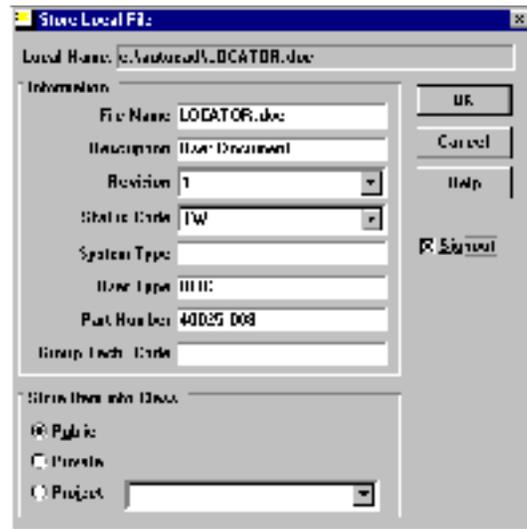
2. Select the required item type. Select Local File unless you have downloaded other rulebase software.
3. Click OK. The Select a File to Store dialog box appears.



In this box you can browse the items in your local file systems in order to make a selection of items to store.

4. Select or type in the name of the file or part to be stored in the Vault.
5. Select or type in the path name of the file or part.

6. Click OK. The Store Local File dialog box appears.



7. Fill in the following information about the file:
  - File Name: Specifies the name of the file. The maximum length can be 80 characters. The Vault name must not already exist in the Vault. The default is the local file name.
  - Description: Specifies the description of the file. The maximum length can be 25 characters.
  - Revision: Specifies the revision number of the file. The applicable revision codes are determined by the file classification. If this field is blank, Locator assigns the default value of the first revision code.
  - Status Code: Specifies the status code for the stored file. If this box is blank, then Locator uses the default of the initial in-work status code.
  - System Type: Specifies the system type. The maximum length can be 8 alphanumeric characters.
  - User Type: Specifies the user type. The maximum length can be 8 alphanumeric characters.
  - Part Number: Specifies the part number. The maximum length can be 20 alphanumeric characters.
  - Group Technology Code: Specifies group technology code (GTC). The maximum length can be 32 alphanumeric characters.

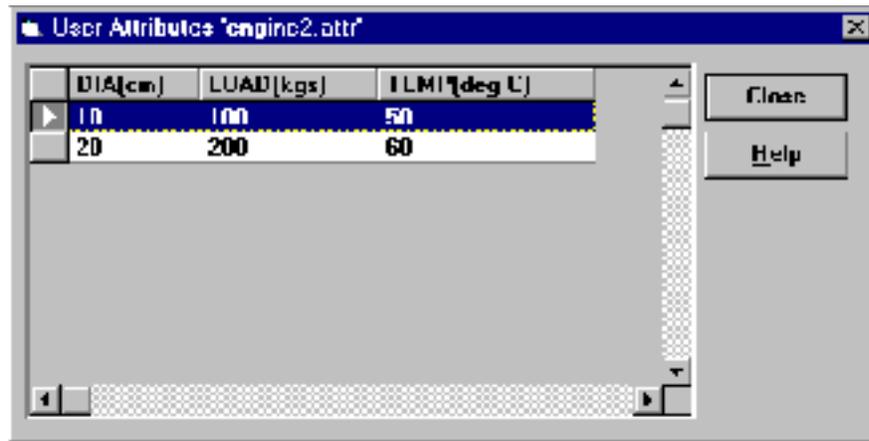
Please note: You can modify the labels of the user-definable fields by modifying the EDMWIN.INI file. See “Parameters Set within Locator” on page 5-4.

8. Select the Vault storage class for the file. The default is Public. If you choose Project, fill in the Project Name box.

### Warning

Files that you store from your PC to the Vault are by default deleted from your directories as part of the Store operation. You can keep the local copy of the file by turning on the 'Signout' check box. On exercising this option, the stored file/part is signed out to you. You must Reset or Checkin the file/part to allow other users to modify it. You can also change the default setting.

9. Click OK. The User Attributes dialog box appears if the file or part to be stored satisfies the rules defined by the system administrator. You can change the values of the attributes. For more information on units of measure refer to "Using Units of Measure" on page 2-10.



## Reserve Name



1. Click Reserve Name on the Optegra Locator toolbar. The Reserve File Name dialog box appears.

A screenshot of the 'Reserve File Name' dialog box. The dialog has a title bar with the text 'Reserve File Name' and a close button. It is divided into two main sections: 'Information' and 'Classification'. The 'Information' section contains several fields: 'File Name' (LOCATOR 3.0.1.doc), 'Description' (User Document), 'Revision' (1), 'Status Code' (1W), 'System Type' (empty), 'User Type' (DDC), 'Part Number' (40025-008), and 'Group Tech Code' (empty). To the right of these fields are three buttons: 'OK', 'Cancel', and 'Help'. Below the 'Information' section is the 'Classification' section, which has three radio buttons: 'Public' (selected), 'Private', and 'Project'. There is also a dropdown menu next to the 'Project' radio button. At the bottom right of the dialog is a checkbox labeled 'Signout' which is currently unchecked.

2. In the File Name field, enter a name as a placeholder for the file. Other information is optional. Choose Signout to check out the file. Enter user-defined attributes and physical data later.
3. Click OK.

Information:

- To remove a reserved name, choose the Reset Item. To enter the data file for a reserved file, use the Store Item or Check In Item options.
- You can reserve a name for a future file only. You cannot reserve a name for a part, file set, or a binder. The name creates a placeholder for a file and not an actual file.

# Copy Item



To make a copy of a file or part, perform the following steps:

1. Select an existing item in the Select Items to Copy window.
2. Click Copy an Item on the Optegra Locator toolbar. The Copy an Item dialog box appears.

**Copy an Item**

Source Name: FILE001.TXT  
Source Rev:

**Copy To**

Copy to Name: FILE.TXT  
Revision: 3 of 1  
status\_code:   
Password:

**Copy Item into Class**

Public  
 Private  
 Project

Buttons: OK, Cancel, Skip, Help

3. Type in applicable information into the text boxes. If you enter an existing name in the Copy to Name field, enter a new revision and status code (if the existing status level is released). You can change the classification or change the password if required.
4. Click OK.

Information:

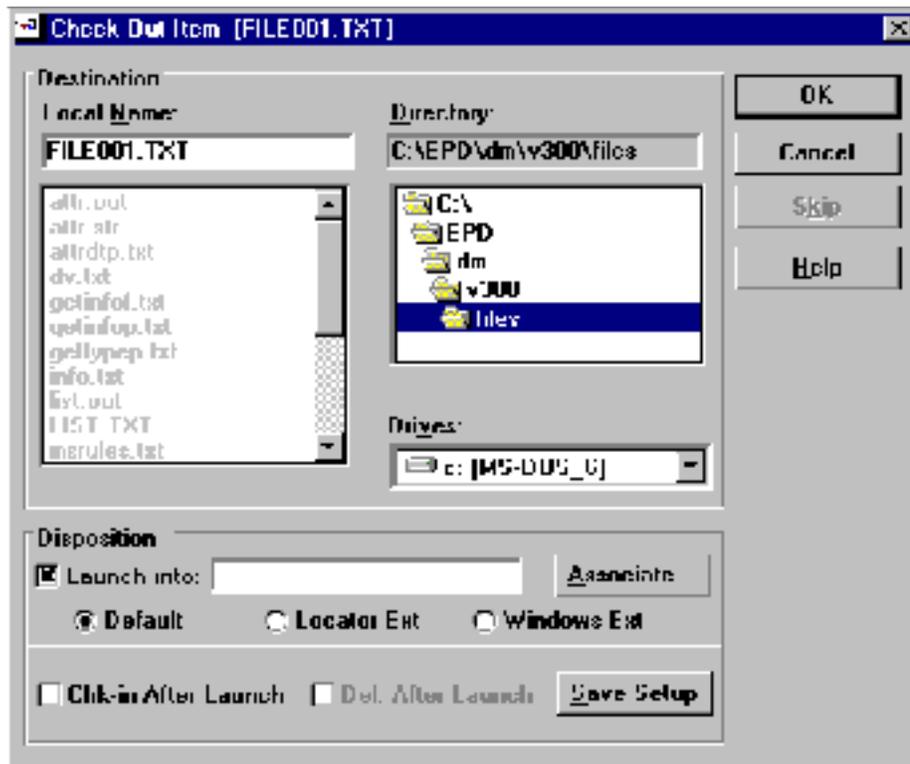
- To create a Vault copy of an item, use the Copy Item option.
- You cannot copy file sets or binders.

# Check Out Item



To check an item out, perform the following steps:

1. Click Check Out Item on the Optegra Locator toolbar. The Select Items to Check Out window appears.
2. Select the items to check out and click OK. The Check Out Item dialog box appears.



If you want the item to open up in its native application when checked out, you can specify this in the Disposition text boxes.

Please note: To configure the Check Out Item dialog box to display the Disposition fields, in the Optegra Locator Log window select Options > Check-Out > Verify Disposition.

3. You can change the parameters in the Check Out dialog box. Choose Save Setup if you want these parameters to be saved for future use.
4. Click OK.

Please note: If you start into an application having selected the Check-In After Launch option, a panel appears when you return to Locator, allowing you to check in the file or part.

Information:

- Check Out Item retrieves a copy of a file, part or file set from the Vault to your PC, allowing you to make changes to it. The system retains the original in the Vault for safekeeping, but now it is locked. A locked file is unavailable to any other user for modification.
- By default Locator starts the application associated with your item. If you only want to retrieve the item and do not want to start up the application, clear the Launch into check box.
- When you have an item checked out from the vault, no one else can check it out or modify it.

Tips:

- To undo a check out operation, use the Reset Item option.
- To save the file, use the Update Item option or Check In Item option. The Check In option also unlocks the file for modification

## Check In Item



The Check In Item option returns an item to the vault, saving the changes made while the item was checked out. After you return an item, another user can check it out.

When you save an item with a status level of Released, Locator creates a new item at the next revision.

To check an item into the Vault, perform the following steps:

1. Click Check In Item on the Optegra Locator toolbar. The Select Items to Check In Window appears.
2. Select the item to be returned.

If user-defined attributes are associated with a file, part, or binder, the User-defined Attributes dialog box appears with the applicable attributes and their current values. Otherwise, go to step 6.

To add or modify the value of an attribute, perform the following steps:

3. Click the value that you want to add or modify.
  - Insert Value inserts a blank entry where the selection is located.
  - Delete Value deletes the entry currently selected.
4. Type the value into the text box.
5. Press the Enter key to enter the attribute value.
6. Click OK when you have the proper attribute values. If the file or part to be checked in has user defined attributes, the User Attributes dialog box appears as shown on page 3-15. You can change the values of the attributes. For more information on units of measure refer to “Using Units of Measure” on page 2-10.

Information:

To transfer your updated version back to Vault without signing it back in, use the Update command. To sign the file back into Vault without updating it with changes made to your local version, use the Reset command.

## Download Copy of Item



Follow these steps to download a copy of one or more items from the Vault:

1. Click Download Copy of Item on the Optegra Locator toolbar. The Select Items to Download window appears.
2. Select the items from the dialog box, and click OK. The Download Copy of Item dialog box appears.

If applicable, a dialog box appears, asking for the password.

You can change the parameters, if required, as described in “Customizing Features for Download and Check Out” on page 2-14.

Enter the password and click OK to complete the download procedure.

Information:

- The Download Copy of Item option creates a copy of each selected Vault item on your PC. You can modify the copy on your PC, but the original remains unchanged in the Vault. To change an item in the Vault, you must first check out that item.
- By default Locator starts the application that your item was created in. If you only want to retrieve the item and do not want to start up the application, unselect the Launch Into check box.

# Update Item



To use the Update command perform the following steps:

1. Click Update Item icon on the Optegra Locator toolbar. The Select Items to Update window appears.

2. Select the items to be updated.

If you try to update a file or part that has not been changed, the system displays a confirmation panel.

If user-defined attributes are associated with a file, part, or binder, the User-defined Attributes dialog box appears, with the applicable attributes and their current values. Otherwise, go to step #6.

To add or modify the value of an attribute, perform the following steps:

3. Click the value that you want to add or modify.
4. Type the value into the text box below.
5. Press the Enter key to enter the attribute value.

The Insert Value command inserts a blank entry where the selection is located. The Delete Value command deletes the entry currently selected.

6. Click OK when you have the proper attribute values. If the file or part to be updated has user defined attributes, the User Attributes dialog box appears as shown on page 3-15. You can change the values of the attributes. For more information on units of measure refer to “Using Units of Measure” on page 2-10.

Information:

- The Update Item option lets you save the changes you have made to a checked-out item (file, part, file set or binder) without having to check it back into the Vault. The item remains checked out to you. No one else can modify it. However, anyone who downloads a copy of the item from the Vault can see the latest changes you have made.
- You may have to close the file(s) from the application as some applications do not allow files to be copied when they are in use.

# Change Info about Item



To change the attributes associated to a file or part follow these steps:

1. Click Change Info About Item on the Optegra Locator toolbar. The Select Items to Change Info window appears.

2. Select an item from the browser and click OK.

If applicable, a dialog box appears, asking for the password.

3. Enter the correct password and click OK.

The Change Info about Item dialog box appears, allowing you to change five different types of information: the file attributes, revision, classification, password, and status level.

Please note: The Change Info dialog box for a Locator part has one additional label, Part Type, which cannot be changed.

4. Click each item that you want to change, and type the new information in the text box. Then, click the respective Change button to adopt the modification. For example, click Change after you have entered a new status code.
  - To change Classification information, first select the new class. If you select Project, you must enter the name of the project. After you have selected a classification, click the Change button in the Classification area to make the change.
  - When changing Status, you can add a comment as to why the status level was changed. The comment can be up to 240 characters. The comments are available in the Status\_History table.

Information:

- You can modify the password, attributes, revision, classification, and so forth
- The Change Info about Item allows you to change the in-built attributes of a file or part, but does not apply to custom, user-defined attributes. To change user-defined attributes, you must use the Update Item or Check In Item options on a checked-out file or part.
- You can change the metadata for a file or part. This function changes all member files of parts, and can be successful only if all those files are available, that is, unlocked.

## Reset Item



To reset a checked-out item, perform the following steps:

1. Click Reset Item on the Optegra Locator toolbar.
2. Select an item from the Select Items to Reset Browser.
3. Click OK.

Information:

- Reset returns a checked out file, part, or file set to the Vault without updating it with any changes. Use Reset Item if you decide not to keep changes you have made to an item since you checked it out. It is also useful if for some reason you lose all your data. Only the user who checked out an item or the system administrator can reset it.
- The reset item is available for use by other users. Any changes you made while the item was checked out are in your local copy only and not made in the Vault version of the file.

# Purge Item

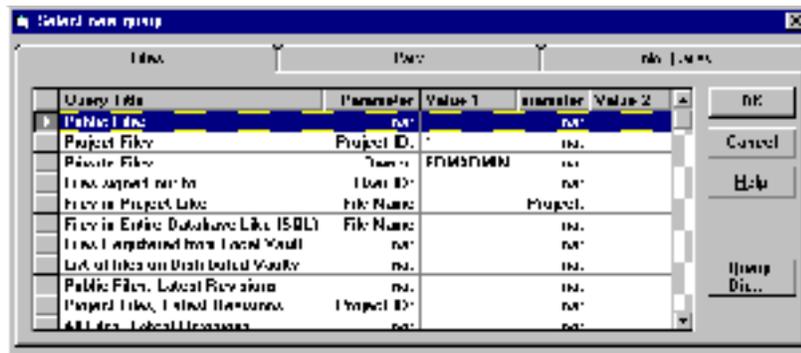


With the Purge Item option you can delete a file or part immediately rather than marking it for deletion at another time. You cannot, however, use Purge to delete file sets or binders.

Please note: You cannot use the File Set Browser to purge items. An item cannot be purged so long as it belongs to a file set. It must first be removed.

To delete a file or part, perform the following steps:

1. Click Purge Item on the Optegra Locator toolbar. The Select Items to Purge Vault window appears.



2. Select the items that you want to purge and click OK.
3. If a dialog box appears and asks for the password, type the password and click OK.

Information:

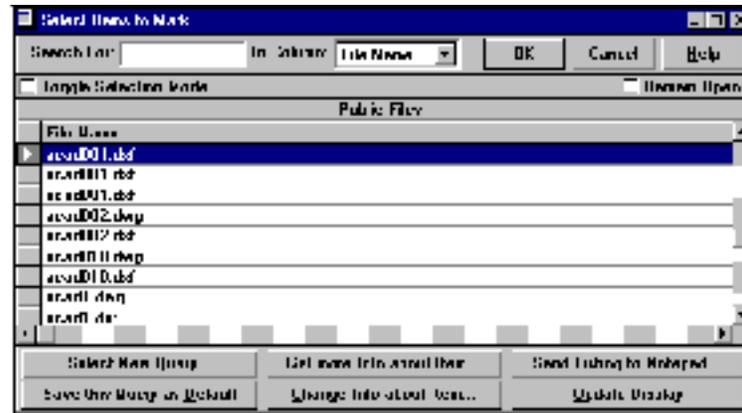
After a file or part has been deleted, it can be recovered only by a System Administrator.

# Mark Item



To mark item files and parts for future system operations, perform the following steps:

1. Click Mark Item on the Optegra Locator toolbar. The Select Items to Mark window appears.



2. Select an item(s) that you want to mark.
3. Select the appropriate field in the File Revision column. If you leave it blank, then the current revision is marked.
4. Type in the password if the item has one.
5. Choose the type of mark:
  - Archive: Preserves the information on tape. A file or part can only be read after marking for archiving. Only the members of a file set or binder can be marked for archiving. You should remove empty file sets or binders after using this option.
  - Restore From Archive: Returns a file or part from the Vault tape archive to active use (on disk storage).
  - Delete: Marks files or parts for deletion. A file or part marked for deletion cannot be accessed (except with the Unmark). Only the members of a file set or binder can be marked for deletion. You should remove empty file sets or binders after using this option.
6. Click OK.

To undo a mark use the Unmark as described in the section “Unmark Item” on page 3-29. If a file or part has been deleted, it can be recovered by a system administrator.

Information:

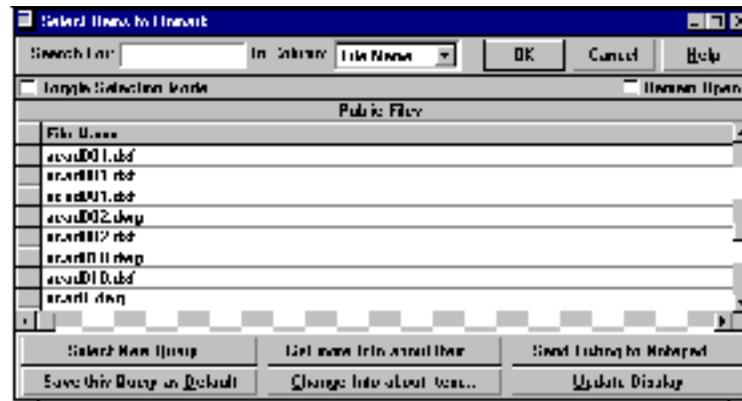
- Marking is the first step in the archive, restore from archive, and delete operations on Vault items. These operations are performed later by a Vault System Administrator and require the use of the magnetic tape drive for the Vault. Two-steps also allows users to change their minds with the Unmark.
- A mark attribute applies only to files and parts. A single mark command is used for all three mark operations, since a file or part can only have one of these three operations applied to it. You can mark files and parts individually or as members of a file set.

# Unmark Item



To undo a mark on an item, perform the following steps:

1. Click Unmark Item on the Optegra Locator toolbar. The Select Items to Unmark default browser appears.



2. Select the item(s) to be unmarked.
3. Click OK.
4. If the file or part has a Password, type in the password.
5. Select the type of mark you want to undo.
  - Archive: Removes a mark for archiving.
  - Restore From Archive: Removes a mark for restoring from the Vault tape archive. If needed, unmark applies to a partially restored selection.
  - Delete: Removes a mark for deletion.
6. Click OK.

Information:

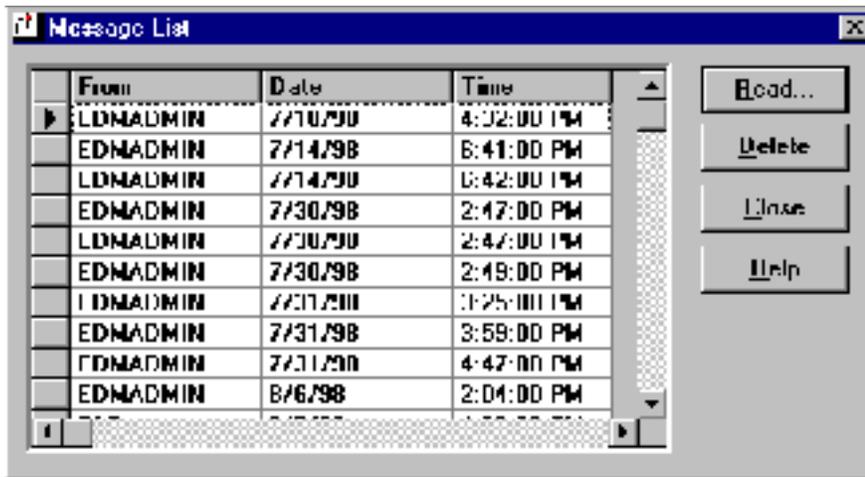
The Unmark allows you to undo the mark for the specified file, part or file set. See “Mark Item” on page 3-27. To determine if a file or part has a mark, use the Mark Item option from the Optegra Locator toolbar or the Vault Browser, which shows the system code, which includes user marks.

# Read Messages

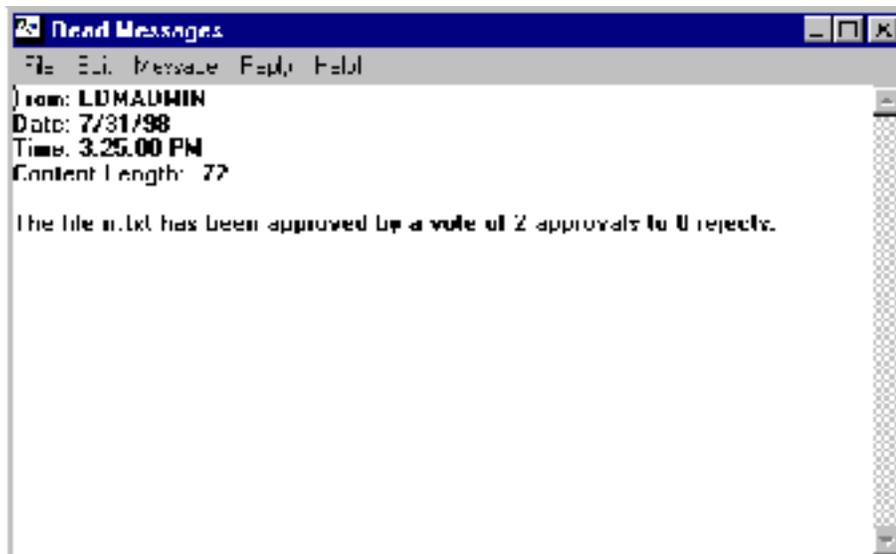


When you have one or more messages, the system notifies you of how many messages you have and asks whether you want to read them. Use the following procedure to read your messages:

1. Choose Yes to the question about reading your messages. The Message List window appears with the list of messages in your Vault mailbox.



2. Select a message and then choose Read Messages. The message appears in the Read Messages window.



**3.** Take care of your messages according to the options in the following table: File:**Table 3-1**

<b>File</b>	<b>Edit</b>	<b>Message</b>	<b>Reply</b>
Save As	Copy	Previous Message	To Sender
Print Setup	Find	Next message	To Sender, include
Print	Find Next	Forward	
Close		Delete	

You can save the message to a file, select and set up the printer you want to use, print the message, or exit from the Read Messages window. The Save As option displays the Save Message File dialog box. Here you can specify a name for the message file, a path name including drive and directory, and a file type. The Print Setup option displays a standard print dialog box.

- **Edit:** You can copy the selected message text to the clipboard, perform a keyword search in either the current message or all messages, or find the next instance of the keyword.
- **Message:** You can display the previous or next message in your mailbox, forward a copy of the message with your comments to anyone in the Vault system, or delete the current message from your mailbox.
- **Reply:** You can display the Send Message with the sender's user ID. Here you can reply to the sender of the current message, with the option of including a copy of the current message. Further details are in "Send Message" on page 3-33.
- **Delete:** You can delete the message from your mailbox and the Vault Message List (unless the message is a request-for-review message from Locator).

**4.** To return to the Message list window, choose File > Close.

## Information:

- Locator allows you to communicate with any other user on your Vault via electronic messages.
- Messages from *RSVP* identify messages that are sent automatically by the Vault.
- For files in review, the Vault automatically sends three kinds of messages by using Locator:
  - Notification messages announce the start of a review.

- Review messages notify reviewers of responsibilities for reviewing a particular file, part, or file set. (Review comments for a review vote are available from the Query commands of Locator.)
- Results messages provide reviewers with the results of a review.

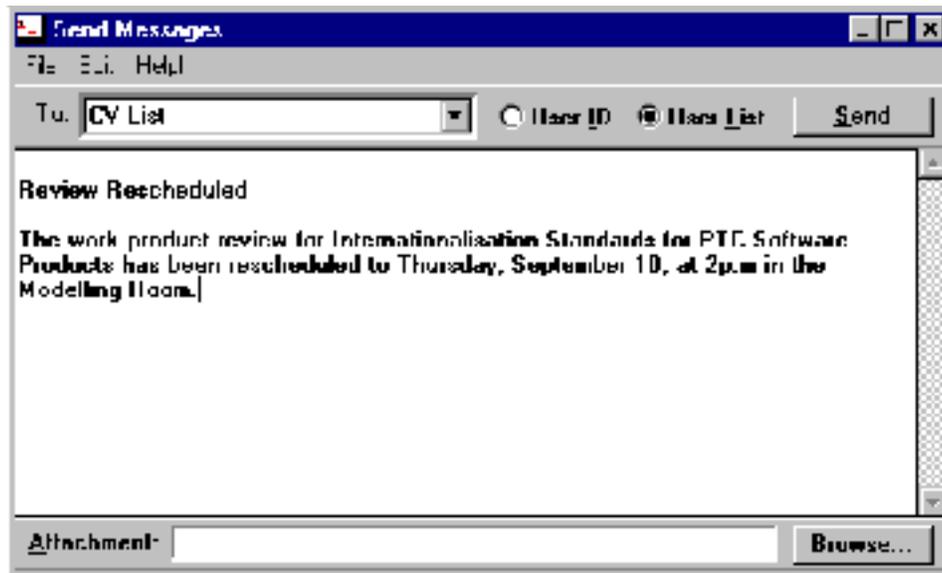
## Send Message



You can send a message to any other Vault users.

1. Click Send Message on the Optegra Locator toolbar

Please note: On the Read Message window you can choose Reply and the Send Message dialog box appears.



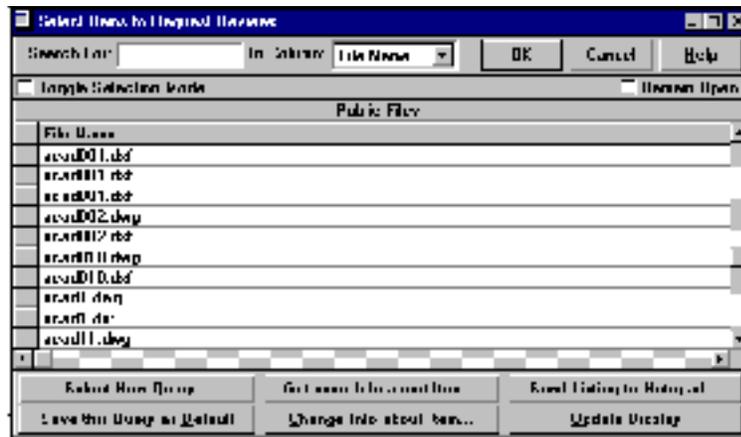
2. Choose User ID or User List as the Destination Type for the message. Click the drop down menu button next to the To: field to see a list of available choices in the specified category.
3. Enter the text of the message. You can also select File > Insert File to copy the contents of a file into the message text window.
4. To attach a file to the associated message, type its full path name into the Attachment box or use Browse to identify the file.
5. Click Send to send the message.

# Request Review of Item



To request a review of a file, part, or file set, perform the following steps:

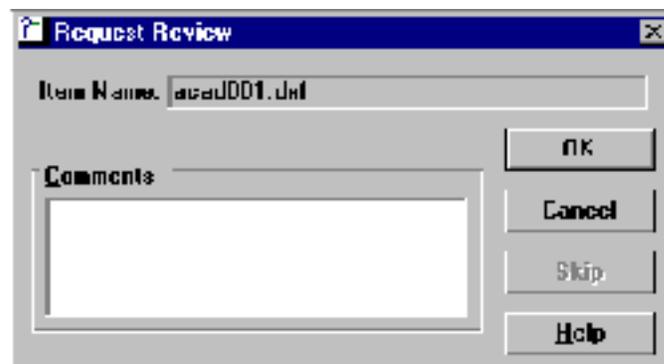
1. Click Request Review on the Optegra Locator toolbar. The Select Items to Request Reviews window appears.



2. Select item(s) for which to request a review. (The previous window is the default.

This marks an item for review. The appropriate people to review the selected item(s) are notified by the system administrator.

3. Click OK. The Request a Review dialog box appears.



4. Type any comments to accompany the review request in the Comments text box.
5. Click OK to send a review copy of the file and your comments to members of the Review List.

Information:

With Locator you can initiate a review of a file, part, or file set. Authorized reviewers can vote to approve or disapprove the item. If approved, the file(s) or part(s) then moves to the next status level.

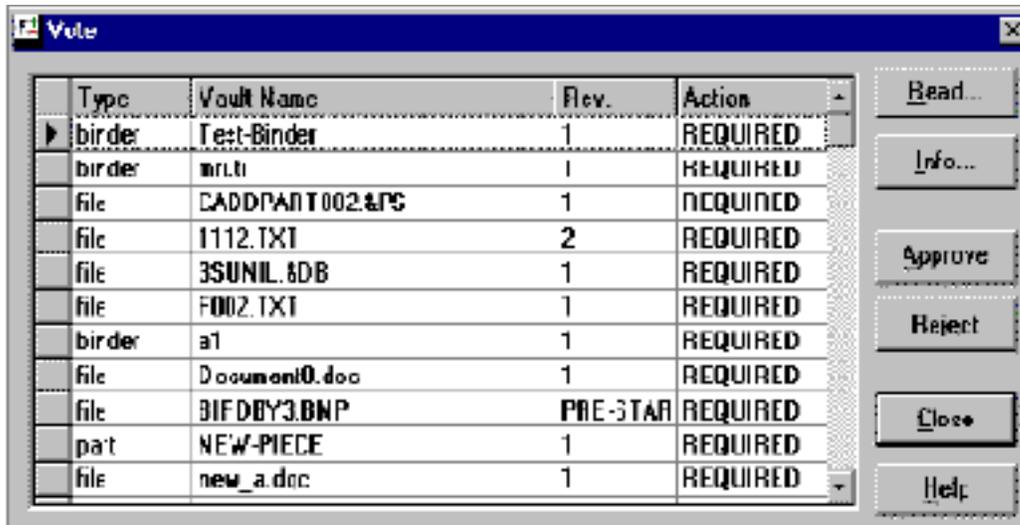
Please note: Review procedures are specified by your Locator System Administrator.

## Vote for/against Item



You can vote to approve or disapprove a Vault item (part, file, file set, or binder) sent to you for review by performing the following steps:

1. Click Vote for/against Item on the Optegra Locator toolbar. The Review List dialog box appears showing the items sent to you for review.



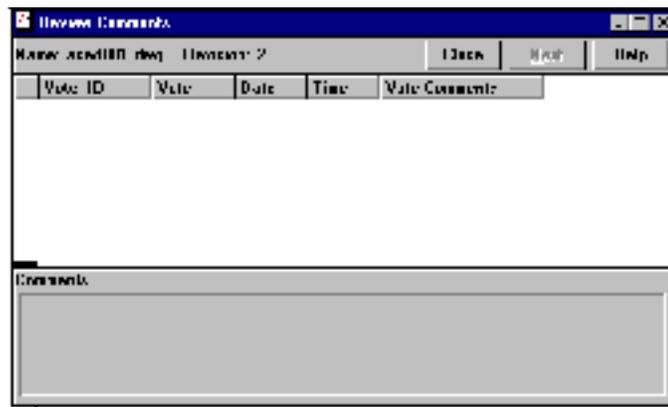
2. Select an item to review.
3. Choose Read or Info as needed.
4. Choose Approve or Reject to vote on the item you have selected.
5. Choose Close to make the vote take affect. The name of the item on which you voted disappears from the list.

# Review Comments



To look at review comments from a completed review of a file or part, use these steps:

1. Click Review Comments on the Locator toolbar. The Select Items to Review Comments Browser appears.
2. Select the appropriate file or part and click OK. The Review Comments window appears with listings, if any, for your selections.



3. To view the review comments for another selected item, click Next.  
Choose Close when you have finished looking at the review.

Information:

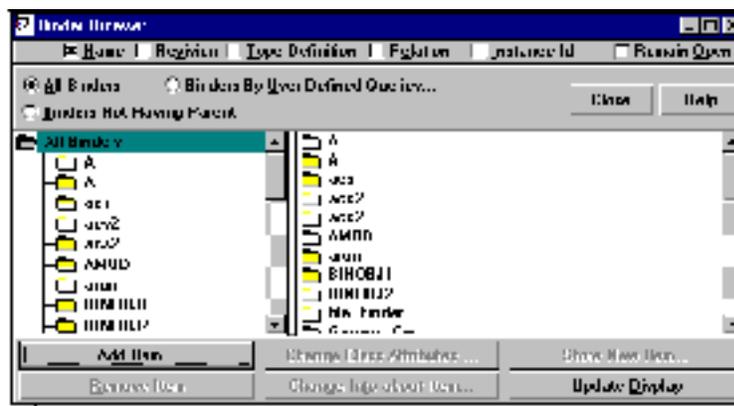
To see the review comments for items still in review, use the queries Review Comments for a File being Reviewed or Review Comments for a Part being Reviewed and select the file or part.

## Binder Browser



With the Binder Browser you can query the Vault for information about binders and display the data. You can also create new binders, change their attributes, and expand them to see their contents. You can use the following toolbar options on a binder: Check In Item, Check Out Item, Download Item, and Update Item.

Please note: Before you use binders, you must set your default browser on the Optegra Locator Log by selecting Browser > Binder Browser. The Binder Browser does not support binder administration.



The Binder Browser, similar to the File Set Browser, displays binders in a tree structure. The All Binders folder branches to individual binders in the Vault. A binder can have files, parts, and other binders as members. If you click a binder, its members appear on the right side of the window.

Information:

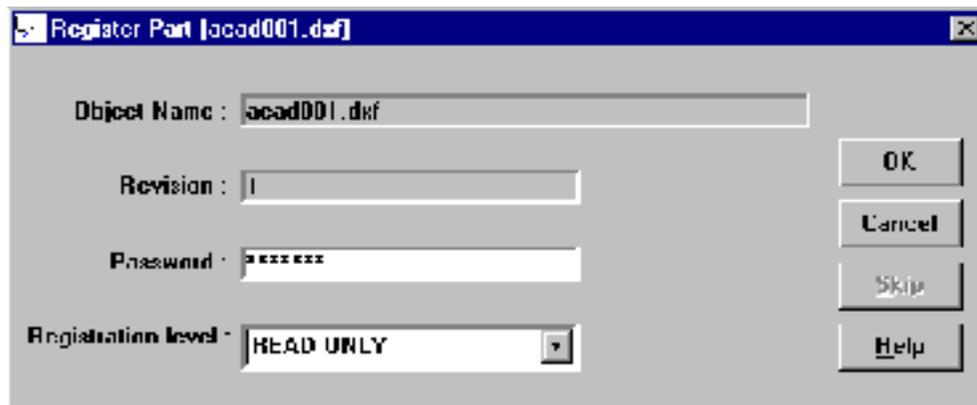
- To display the Attribute Name, Revision, Type Definition, and Relation and Instance Id, select the appropriate check boxes.
- With the Binders By User Defined Queries option, you can selectively display the binder list.
- If you select Binders Not Having Parent option, the Binder Browser lists all the root binders that are not the child of any other binder.
- You can select multiple items from the Binder Browser by dragging the mouse across several items.
- Like other browsers, you can invoke the Binder Browser in two ways: as an independent tool or as a browser called from another tool like Check In Items.

## Register Item



To register an item in the Distributed Objects Directory, follow this procedure:

1. Click Register Item on the Optegra Locator toolbar. The Select Items Browser with the list of files from local Vault appears.
2. Select the item you want to register and click OK. The Register Part dialog box appears.



The Object Name and Revision of the selected object appear in the respective text boxes.

3. Enter a password, if required.
4. Select the Registration level from the list.
5. Click OK to register the object.

Information:

In the Distributed Vault environment, you can register objects for direct access from other vaults. The level of registration determines the accessibility.

## Unregister Item



To remove an item from the Distributed Vault, you must unregister it following these steps:

1. Click Unregister Item on the Optegra Locator toolbar.

The Select Items Browser with the listing of all the registered items appears.

2. Select the items you want to unregister.
3. Click OK. The Unregister Object dialog box appears.

The Object Name and Revision Number of the object to be unregistered appears in the respective text boxes.

4. Enter a password, if required.
5. Click OK.

The object is processed for unregistering and the registration level of the object is changed to local.

Information:

When you unregister an item, the process removes the item from the Distributed Vault. Therefore, you cannot access the object from the Distributed Objects Directory (DOD).

# Add Notification



To add a notification, follow these steps:

1. Click Add Notification on the Optegra Locator toolbar. The Select Items to Notify Browser appears.
2. Select the File(s) and Part(s) from the list.
3. Click OK. The Add Notification dialog box appears.

**Notify part [acad001.dxf]**

Notification Name : REVISION CHANGE

Object Name : acad001.dxf

Object Revision : 1

Event Name : CHANGE\_OBJECT\_PRIMARY

**Notification Information**

Message to User       Message to User List

Message to : GLB

Notify mode : NO MESSAGE

**Optional Information**

Expiration Date - 8/15/98  
(M/d/yy)

Expiration Time : 10:30:00  
||

Expiration Count : 3

Buttons: OK, Cancel, Skip, Help

4. Enter a name in the Notification Name text box.
5. In the Event Name box select an event you want to use, for example, CHANGE\_OBJECT\_REVISION.
6. Select Message to User or Message to User List and enter the information in the text box.
7. In the Notify Mode box, enter the mode to identify under what conditions that you want the notification sent, for example, BOTH SUCCESS AND FAILURE.

8. If required, type the expiration date, expiration time, and expiration count in the respective text boxes. This information is optional.
9. Click OK.

The object is processed for adding the notification.

### Warning

Add Notification supports only the actions of sending a message to a user or a user list. It does not support action taken while importing or exporting objects.

### Information:

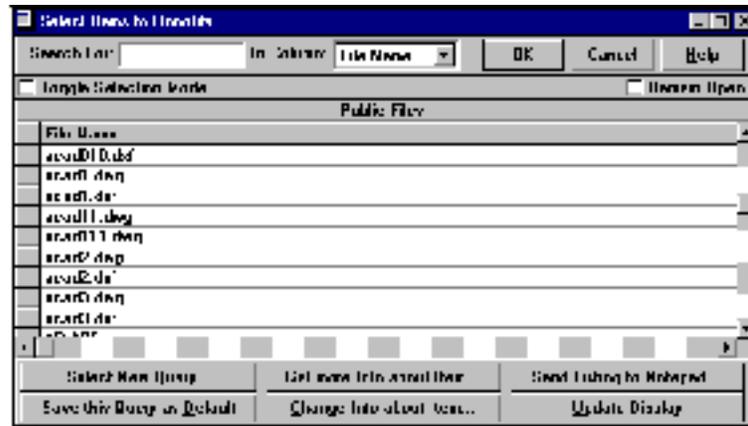
Add Notification establishes a relationship between an event and an action. Use it to monitor changes in the database. Your notification can include an action template for notifying a user or a user list.

# Delete Notification



To delete a notification follow these steps:

1. Click Delete Notification on the Optegra Locator toolbar. The Select Items Browser appears with a list of all the notifications owned by the user currently signed on.



2. Select the notifications you want to delete.
3. Click OK. The selected notifications are deleted.



# Working with Binders

---

The chapter contains the following information about working with binders:

- Understanding Binder Concepts
- Setting Up and Using Binders
- Selecting a User-defined Query
- Integrating the Binder Browser

# Understanding Binder Concepts

A Binder is a container for logically related objects. You can use a binder to collect parts, files, or other binders. You create the container and then add objects to it by relating the objects to the binder. The entire collection is referred to as “the binder” after objects have been connected to the container.

After you create a binder, you menu option access all objects in the binder. When one of the binder objects changes, you can easily determine which other objects are affected. Binders can have their own revision and status levels.

Binders can be used for the following purposes:

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

You can perform the following operations on binders from the Locator toolbar:

- Check Out Item
- Check In Item
- Download Copy of Item
- Update Item

## Relation

A Relation (or relationship) is the name of the connector between the container (binder) and the object (file or binder) it contains. Each relationship is defined by the specific parent and child that it relates to. The same parent and child can be related more than once. In this case, you have multiple instances of the same relation. You can select two types of relationships for your binder: aggregation and association. Your choice is based on how you want to do check outs and reviews.

- Aggregation is used to manage a collection of objects that are dependent on each other. In an aggregation relationship, a change to one member of the collection affects the other members. You use aggregation when you want constraints and dependencies between members.
- Association is used to manage a group of objects that are independent of each other or that contain different pieces of a large project that 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.

## Setting Up and Using Binders

You can access the following options from the Binder Browser. The procedures are explained in the next sections.

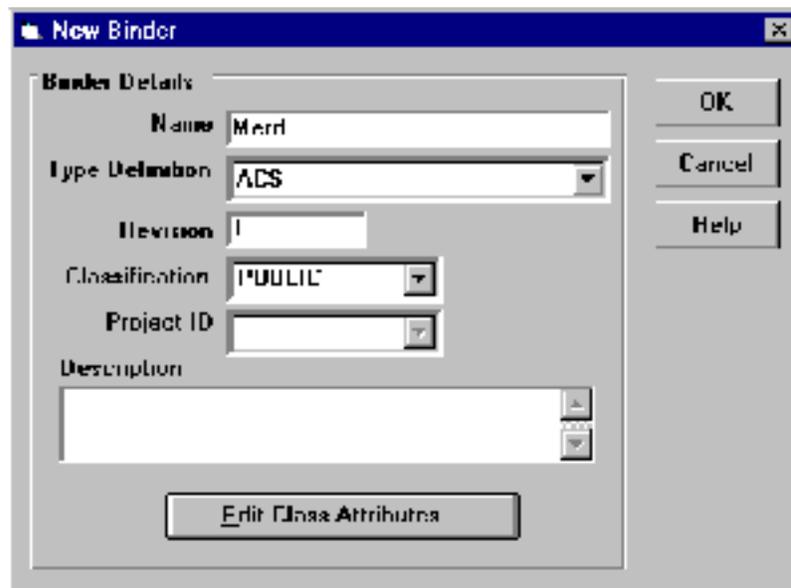
- Add Item
- Change Class Attributes
- Store New Item
- Remove Item
- Change Info about Item
- Update Display

### Adding an Item

You can create a new binder or add a member to an existing binder, using the Add Item option in the Binder Browser. To create a new binder, select All Binders at the top of the tree structure. To add a member, select the name of the receiving binder further down in the structure.

To create a new binder, perform the following steps:

1. Select All Binders from the Binder Browser.
2. Click Add Item. The New Binder dialog box appears.



The image shows a 'New Binder' dialog box with the following fields and controls:

- Name:** Merri
- Type Definition:** ACS
- Revision:** 1
- Classification:** PUBLIC
- Project ID:** (empty)
- Description:** (empty text area)
- Buttons:** OK, Cancel, Help, and Edit Class Attributes.

3. Enter the name for your new binder and other information as follows:
  - **Type Definition:** Specifies the type definition of the new binder. Select from the list of predefined type definitions. The attribute groups attached to the selected type definition apply to the new binder to be created.  
  
You must enter Type Definition as new binders without type definition cannot exist.
  - **Revision:** Specifies the revision number of the binder. The applicable revision codes are determined by the binder classification. If this field is blank Locator assigns the default value of the first revision code.
  - **Classification:** Specifies the Vault storage class for the binder, either Public (default), Project, or Private. **Project ID:** You must enter the Project ID, if the binder classification is project.
  - **Description:** Specifies the description for binder. The maximum length can be 250 characters. This is optional.
4. Click Edit Class Attributes option to assign values to the attributes defined for the related type definition. Follow the procedure specified in the next section “Specifying Binder Class Attribute Values” on page 4-4. This is optional if there is no mandatory attribute group defined for the related type definition.

Please note: All the binders of the same type definition have the same attributes. However, the attribute values are specific to each individual binder.

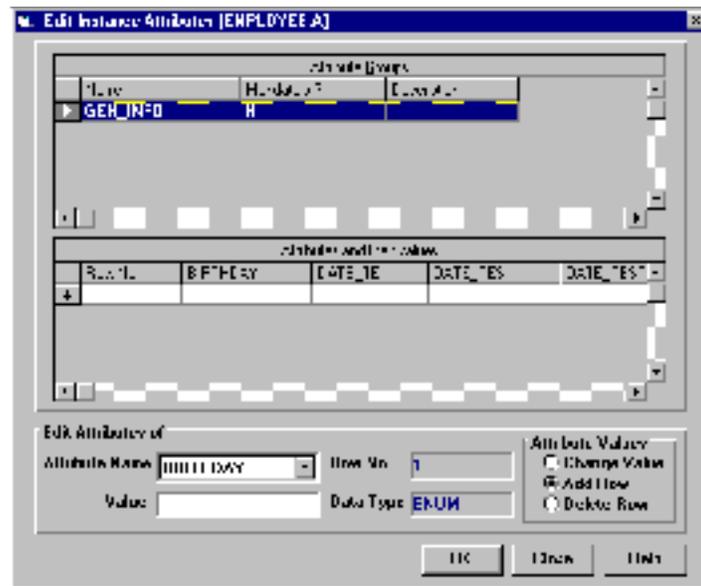
5. Choose OK to confirm your entry and close the dialog box.

## Specifying Binder Class Attribute Values

An instance of a binder is created when a new item with the base class Binder is created from the type definition. The attribute groups created for the type definition apply to the instance of the binder. The values given to the attributes are unique to an instance of a binder.

To specify binder class attribute values, perform the following steps:

1. Display the Add Binder dialog box if it is not already displayed.
2. Select Edit Class Attributes. The Edit Class Attributes<typedef,binder name> dialog box appears.



Attribute Groups:

Name: Specifies the name of the attribute group.

Mandatory: Specifies that the attribute is mandatory. The value can be either Y (yes) or N (no). If Y, then at least one row of attribute values is required.

Description: Specifies the description of the attribute group.

Row No: Specifies the row number of the attributes belonging to the selected attribute group.

3. Select the attribute group that you want to modify.

The attributes and the attribute values of the selected attribute group are listed in the display box.

The attribute names of the selected attribute group appear as the column headings and their values are reflected in the rows.

4. To edit the values in a row, select the row you want to change. You can also insert a new set of values in the Attributes and their values display box. To do so, select the row above or below which you want to add the new set of values.
5. Select the attribute name of the attribute you want to edit from the Attribute Name combo box.

The corresponding attribute information appears in the Value, Row No. and the Data Type text boxes.

6. In the Attribute Values box, select one of the operations.
7. Choose OK to confirm your entry and keep the dialog box open.

## Adding a Member to a Binder

To add a member (file or part) to a binder follow these steps:

1. On the Binder Browser, select the parent binder to which you want to add the members from the Vault.
2. Click Add Item. The Vault Browser appears.
3. Select one or more item(s) from the Vault Browser.
4. Click OK.

Please note: You can also drag-and-drop any item, including binders or members of other binders, from the right side of the Binder Browser onto the binders displayed on the left and proceed as follows.

The Add Member to Binder dialog box appears.



**Base Class:** Specifies the base class of the member that can be a file, part or binder.

**Name:** Specifies the name of the member.

Revision: Specify the revision of the member (that is, file, part, or binder).

Instance ID: Displays the member instance ID.

Relation Type: Specifies the relation type. Select either Association or Aggregation as the relation type.

Description: Specifies the description of the member. The maximum length can be 250 characters. This is optional.

Type Definition [Relation]: Specifies the type definition (optional) of the new relation to be added. Select from the list of predefined Relation type definitions. The attribute group attached to the selected type definition applies to the new relation.

Relations without type definitions can be created. Hence it is optional to select type definitions for relations.

5. To enter attribute values for a relation instance, click Edit Relation Instance Attributes and follow the procedure in the next section. Entering attribute values is optional.

Please note: The Edit Relation Instance Attributes button is only available when the parent binder has attribute values.

6. Click OK to confirm your entry and close the dialog box.

## Specifying Relation Attribute Values

When you add a member (file, part or binder) to a binder, an instance of a relation is created. The attribute groups created for the type definition of relation apply to the instance of the relation. The values given for attributes are unique to a particular instance of a relation.

You can create instance attributes when you add or edit members. To edit a relation instance attribute follow these steps:

1. Choose Binder Browser > Add Item if the Display the Add Member to Binder dialog box it is not already displayed.
2. Select Edit Relation Instance Attributes. The Edit Class Attributes dialog box appears.
3. Follow steps 3-7 as described in “Specifying Binder Class Attribute Values” on page 4-4.

## Deleting a Binder or a Binder Member

You can remove a binder and binder member using the Remove Item option from the Binder Browser. If you select All Binders from the tree structure on the left side of the Binder Browser, you can remove a binder. If you select a single binder, you can remove member(s) of the selected binder.

To delete a binder or binder item, perform the following steps:

1. Select one or more binders you want to delete from the left side of the Binder Browser. The selected binder(s) are highlighted.
2. Choose Remove Item. The Delete Binder dialog box appears.



3. Choose Cascade to delete the selected binder and all its related objects.

Please note: Files and parts are not deleted from Vault.

4. Click OK to delete.

Repeat steps 2 through 4 for each binder that you want to delete.

To remove a binder member, perform the following steps:

1. Select one or more binder members you want to delete. The selected member(s) are highlighted.
2. Choose Remove Item. A confirmation message appears for each item that you have selected.
3. Click OK to delete the member.

## Assigning Class Attributes

Attribute values can be assigned for binders and binder relations. Attribute groups are defined for the type definition of an object, that is a binder or a relation. An attribute group can have any number of attributes. Attributes of groups that have the same type definition must be the same, whereas the attribute values are specific to an object.

## Changing Attributes of a Binder

You can change, add, or delete the class attribute values of the selected binder using Change Class Attributes.

To change the class attributes of a binder follow this procedure:

1. Select a binder from the left side of the display area in the Binder Browser.
2. Click Change Class Attributes. The Edit Class Attributes <typedef.selected binder name> window appears.
3. To change the class attributes follow the steps described in “Specifying Binder Class Attribute Values” on page 4-4.

## Changing Attributes of a Relation

You can also change, add, or delete the values of the relation instance (that is, a member) using Change Class Attributes.

To change the class attributes of a relation, perform the following steps:

1. Select an object on the right side of the Binder Browser.
2. Click Change Class Attributes. The Edit Class Attributes<relation\_ typedef.member\_name\_in\_parent binder> window appears.

To change the relation attribute values follow the steps described in “Specifying Relation Attribute Values” on page 4-7.

## Changing Information about an Item

You can use Change Info about Item to change various attributes of a binder such as Revision, Classification, Description and Status Level or to edit the description of a relation.

If you select a binder from the left side of the display area, only the binder attributes can be edited. If you select binder members from the right side of the display area, only the description of the member relation can be edited.

## Change Binder Attributes

To change the attributes of binders, perform the following steps:

1. Select a binder whose attributes you want to change.
2. Choose the Change Info about Item button. The Change Info of Binder <selected binder name> dialog box appears.
3. Select each field that you want to change and type in the information. For example, to change the revision, click in the Revision field and type the new revision number.

To change Classification information, first select the new class. Select Project and enter the Project ID. Similarly, you can change the Description and Status Level of the selected binder.

4. Click OK to affect the changes.

## Change Description of Relation

To change the description of relation(s), perform the following steps:

1. Select a binder member whose description path you want to change.
2. Click the Change Info about Item. The Change Description <selected Relation Name> dialog box appears. The selected relation description appears.
3. Enter the description in the Description text box.
4. Click OK to confirm your entry and close the dialog box.

## Storing a New Item

On the Binder Browser, you can choose Store Item option to store a file in the vault and add it to the selected binder.

To store a new item, perform the following steps:

1. Click Store New Item from the Binder Browser.
2. Follow the steps described in the section “Store Item” on page 3-13

## Updating the Display

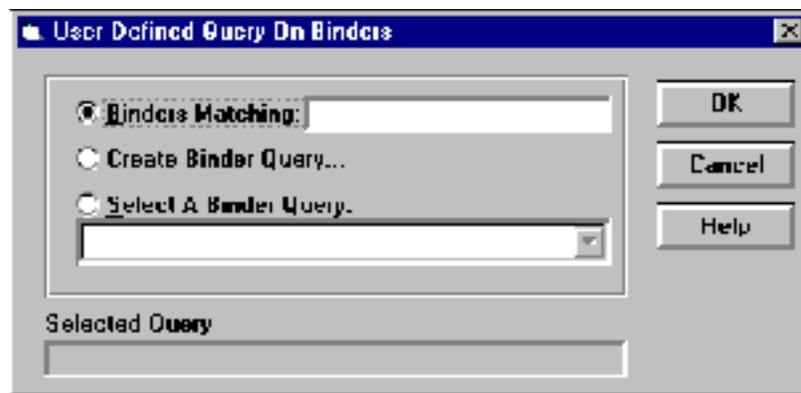
You can update the Binder Browser display by clicking Update Display.

## Selecting a User-defined Query

You can display binders of your choice by using the User Defined Query Option(UDQ). A UDQ is stored in the IQF subdirectory of the Locator with a .hlq extension. The query file basically contains an SQL statement that you can edit later to suit specific requirements. You can also add more conditions to the binder query through a high-level interface.

To select a query follow the procedure given below:

1. Click Binders By User Defined Queries on the Binder Browser. The User Defined Query on Binders dialog box appears.



2. You can choose from three options:

**Binders Matching:** Matches the text string specified in the corresponding text box. All the binders with names beginning with the specified string are displayed.

**Create Binder Query:** Displays the newly created query.

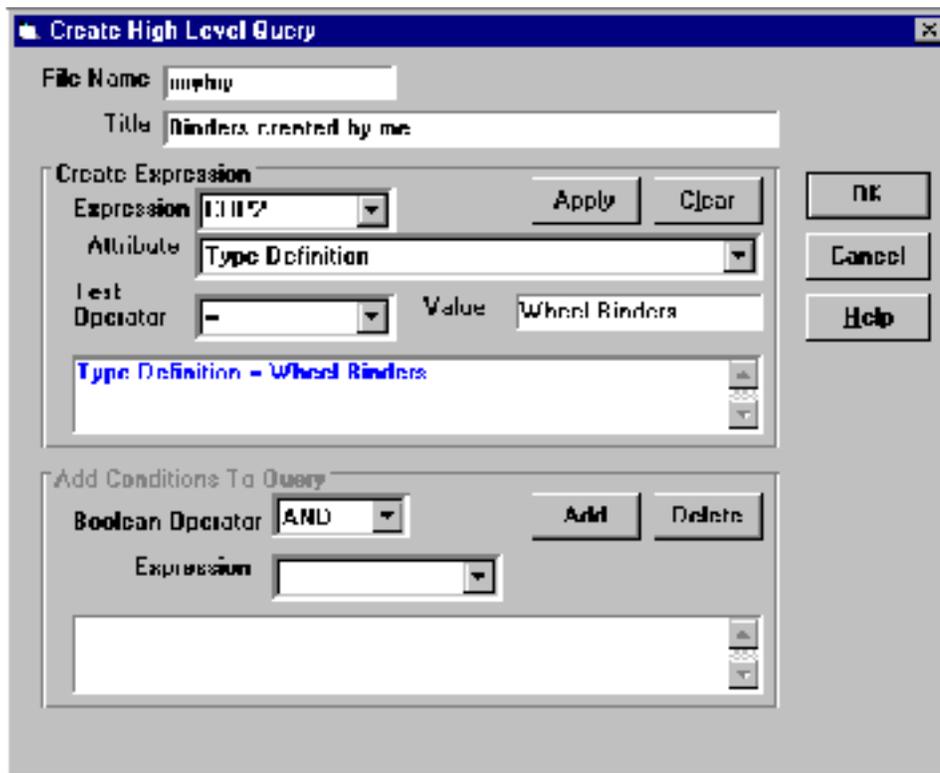
**Select A Binder Query:** Allows you to select one of the existing UDQs.

3. Click OK. The system executes the related query and displays the selected binders on the browser.

## Create Query

The Create Binder Query allows you to create new User Defined Queries (UDQ). Follow the procedure given below to create user defined queries:

1. Select the Create Binder Query option from the User Defined Query in the Binder dialog box. The Create High Level Query dialog box appears.



2. Enter a name in the File Name text box.
3. Enter a title in the Title text box. This is required because the title distinguishes a UDQ from other existing ones.
4. Select an expression by clicking the editor button to the right of the Expression combo box.
5. Enter a value in the Attribute combo box by clicking the editor button to the right of this combo box. This lists a set of all the attributes and field names of Vault tables related to binders, for example, Type Definition and Classification.
6. Click Apply to save the expression. These expressions are used to form conditions for creating a User Defined Query (UDQ).

In the Add Conditions To Query box, you can make conditions by selecting Boolean operators and expressions.

- 7.** Select a Boolean operator by clicking the editor button to the right of the Boolean Operator combo box.
- 8.** Select an expression by clicking the editor button to the right of the Expression combo box.
- 9.** Click Add to append the condition to the existing ones.
- 10.** Click OK. The UDAQ is stored in the IQF subdirectory with the file name that you have entered in the File Name text box.

## Deleting a Condition

To delete a particular condition, perform the following steps:

- 1.** Select the corresponding expression from the text box where the expression is displayed.
- 2.** Click Delete. The selected query is deleted from the IQF subdirectory.

## Integrating the Binder Browser

The Binder Browser can be integrated with other Locator tools in the following ways:

- A browser option in Optegra Log. With the Binder Browser selected, the Binder Browser must appear whenever you choose Check Out Item, Check In Item, Update Item, and Download Copy of Item.
- Provision to invoke the Store Item option. From the main form, Store Item tool can be started to store files and parts, which can be made members of any binder subsequently.
- Invoking Vault Browser. For selecting files and parts to be made members of a binder, you can use the Vault Browser.

This chapter describes the program preferences for Locator and how to configure them in the initialization file. This material is intended for advanced users. The following information is included:

- Configuring the Initialization File
- Locator Directories
- Parameters Set within Locator
- Field Display Definitions

## Configuring the Initialization File

The `EDMWIN.INI` is the initialization file that contains all the configuration parameters for Locator. The `EDMWIN.INI` file resides in your `WINDOWS` directory for Microsoft Windows or `WINNT` directory for Windows NT.

You can set the parameters in two ways:

- Click Save Setup to save any settings in that window.
- Edit the `EDMWIN.INI` file, for example, the `show_log` parameter. You can use any text editor, such as Notepad.

The next table contains all of the configuration parameters that can be set from within Locator with cross-references to their corresponding menu commands.

Initialization parameters are organized into various sections: Locator, Selection, Display(s), Variables, Hosts, Locator-Filemenu(s), Debug, and so forth. The values for all of the parameters in `EDMWIN.INI` and their defaults are listed in this chapter.

# Locator Directories

Locator installs files in two directories:

- The default windows directory (`windows`, `winnt`, or `winnt35`).
- installation directory of the locator software as specified by the user

The installation program copies the initialization files, `edmwin.ini` and `wedmlib.ini`, to the `windows` or `winnt` directory. A set of Dynamically Linked Libraries (DDLs) are copied to the same systems directories to support Locator.

The installation directory is created by the install program on a drive specified by the user (the Locator home drive). Files are stored under the installation directory specified by the user and grouped under the following subdirectories:

- `addin` — Contains sample menu and tool bar configuration information.
- `audit` — Initially empty and used internally.
- `files` — Initially empty. The default location for files transferred from the Vault.
- `iqf` — Contains standard IQF, RQF, and SQF files. The default location for all IQF files.
- `log` — Initially empty. The default location for saved session log files.
- `lrb` — Initially contains the LOCAL interface (`LOCAL.DLL`). Default directory for all local interfaces.
- `mdb` — Vault host and network support files.
- `menus` — Contains sample menu and toolbar configuration information.

Please note: The files created in `audit`, `files` and `log` subdirectories are not deleted. The user can refer them later, if required.

## Parameters Set within Locator

A list of definitions of the parameters in the [edmwin] section of the EDMWIN.INI file follows.

<b>.INI Parameter</b>	<b>Description</b>
lhostname	Vault Hosts default value at startup.
ldomainname	Vault Domains default value at startup.
lusername	Vault User ID default value at startup.
home_directory	The application's home directory.
file_directory	Local File directory for Optegra.
iqf_directory	Interactive Query Facility query files location.
lrb_directory	Local interfaces (rulebases) location. Any manual changes for this token need to be reflected in the WEDMLIB.INI file.
audit_directory	Local audit file location.
mdb_directory	Visual basic screen driver location.
mnu_directory	The Application's menu directory.
list_type	Vault Browser default list type (P, F, S, I).
list_query	Name of the default query file to be executed.
list_params	Default list parameters (if required).
delete_after_store	Remove local item after Store (Y, N).
delete_after_checkin	Remove local item after Check-In (Y, N).
overwrite	Overwrite existing local versions on Check-Out or Download copy of item (Y, N).
log_left	Left starting position for Session Log in pixels.
log_top	Top starting position for the Session Log in pixels.
log_height	The height of the Session Log in pixels.
log_width	The width of the Session Log in pixels.
show_log	Show the Session Log on startup (Y, N).
float_window	Allow the Session Log to move separately from the tool bar (Y, N).
full_trace	Show full trace capability in the Session Log for debugging purposes (Y, N).
confirm	Confirm transactions before execution (Y, N).
passwords	Prompt for item passwords (Y, N).
attributes	Enable attributes (Y, N).
browser	Default browser.

<b>.INI Parameter</b>	<b>Description</b>
list_type_dv	Vault Browser Default list type for DV related queries (P, F, S, I).
list_query_dv	Name of the Default query file to be executed for DV related operations.
list_params_dv	Default list parameters for DV related queries if required.
current_userid	Vault Userid who is currently signed on.
current_hostname	Vault Hostid on which the client is currently signed on.
current_domain_name	Vault Domainid on which the client is currently signed on.
dv	Whether the Vault Host on which client is currently signed on is in DV environment or not (Y,N).
objects	Type of Vault on which the operations are to be performed (local, distributed).
awlang	Language setting (C:/german/japanese/french).
launch_host_app	Launch "Vault Host and Domain Configuration" module when a wrong vault host name is specified during login (Y, N).
launch_login_app	Launch "Optegra Login" module when the user authorization fails during login (Y, N).
launch_mail_app	Launch "Read Messages" module on a successful login (Y, N).

Following are parameters definitions in the [ checkout ] section.

<b>.INI Parameter</b>	<b>Description</b>
follow_thru	Follow through with an application (Y, N).
checkin_after	Check-In after completion of follow through application.
delete_after	Delete the local item after the Check-In is completed.
default_destination	Default location for local items checked out/downloaded from Vault.
extension_type	Specify how local PC file names are to be created from Vault file names.Values for this token follow:  usertype = Use the Vault User Type for the extension of the file.  edmname = Expect to find the extension at the end of the file name (default).
shorten_filenames	Shorten Long file names to 8.3 MS-DOS format(Y, N).
verify_destination	Allow application launch (Y, N).
verify_overwrite	Allow local copy overwrite (Y, N).
verify_disposition	Allow user-specified path for local copy (Y, N).

Following are parameters definitions in the [palette] section.

<b>.INI Parameter</b>	<b>Description</b>
tooltips	Display Tooltips when pointing to an icon (Y, N).
pal_left	Left starting position of Tool Bar in pixels.
pal_top	Top starting position of Tool Bar in pixels.
pal_height	The height of Tool Bar in pixels.
pal_width	The width of Tool Bar in pixels.
animate_icon	Animate icon toggle (Y, N).
statusbar	Status bar toggle (Y, N).

Following are definitions of the parameters in the [query] section.

<b>.INI Parameter</b>	<b>Dialog Control Name</b>
query000	Object Name and extension of IQF execution files used for File category.
query001	Object Name and extension of IQF execution files used for Part category.
query002	Object Name and extension of IQF execution files used for Info Queries category.

Following are parameter definitions in the [store] section.

<b>.INI Parameter</b>	<b>Description</b>
uppercase=y	Set the default character style for storing Vault names to uppercase.

Please note: If neither the Vault EDM.DEFAULTS file nor the user's environment contains the statement EDMCASE=MIXED, then you must add this parameter to the EDMWIN.INI file.

If you experience problems using the Store option, locate the [Store] section of the EDMWIN.INI file in the WINDOWS or WINNT directory. Change the line STORE\_APP=STORE to STORE\_APP=CVSTORE.

## Field Display Definitions

The following list contains parameter definitions in the [field\_width] section of the EDMWIN.INI file. It also contains default values.

```
CLASS=3
OWNER=12
STATUS_CODE=8
SYSTEM_TYPE=8
USER_TYPE=8
PART_NUMBER=20
DESCRIPTION=25
CLASS_CODE=32
SYSTEM_CODE=1
USER_ID=12
NODE_NAME=32
CREATE_DATE=6
CREATE_TIME=6
UPDATE_DATE=6
UPDATE_TIME=6
PART_TYPE=80
FILE_TYPE=80
```

The following list contains parameter definitions in the [field\_title] section of the EDMWIN.INI file. It also contains default values.

```
CLASS=Class
OWNER=Project
STATUS_CODE=Status Code
SYSTEM_TYPE=System Type
USER_TYPE=User Type
PART_NUMBER=Part Number
DESCRIPTION=Description
SYSTEM_CODE=System Code
USER_ID=User ID
NODE_NAME=Node Name
CREATE_CLASS_CODE=Group Tech. Code
DATE=Creation Date
CREATE_TIME=Creation Time
UPDATE_DATE=Update Date
UPDATE_TIME=Update Time
PART_TYPE=Part Type
FILE_TYPE=File Type
```



# Query Facility System

---

This chapter describes the Interactive Query Facility (IQF) for querying the Vault database. The chapter includes information on the following:

- Queries in Locator
- Standard IQFs in Locator
- Overview of IQFs
- IQF Keywords
- Selection Query Files (SQFs)
- Internal Queries (RQFs and HQFs)

## Queries in Locator

Vault provides an Interactive Query Facility (IQF) for querying the Vault database. The IQF is described in detail in the *Vault Interactive Query Facility Guide*. IQF is based on SQL. Every Locator query has at its heart an SQL select statement.

Locator has four types of queries which are distinguished by their file extension:

- IQF Query Files (.IQFs) enable Locator users to perform their own queries of the Vault database. Locator comes with standard queries that you can customize and add your own. IQFs can prompt the user for information in making the query. The results from the query are displayed in a separate window by a separate viewing application.
- Selection Query Files (.SQFs) are used internally to select the view to be displayed in the Vault Browser and the Vault File Set Browsers. SQF files can also be customized by the user.
- Help Query Files (.HQFs) are used by .IQF and .SQF files to provide drop-down lists.
- Reserved Query Files (.RQF) are used internally by Locator. These files should not be modified.

Query files with the IQF extension, by default, use the Locator IQF command. All other types connect directly to Oracle via the PDMDM account.

Locator query files are stored in the IQF default directory. This default directory is %EPD\_HOME%\iqf on the Locator home drive. The path is stored in the `iqf_directory` parameter in the Locator initialization file `EDMWIN.INI`.

## Standard IQFs in Locator

Several IQF query files are shipped with Locator and installed in the IQF default directory. IQF file descriptions follow. If the description requires the user to specify a value, the system prompts the user for the query parameter values.

**Table 6-1 IQF Query File Descriptions**

<b>IQF Query File</b>	<b>Description</b>
ATTR.IQF	User-defined attribute data for a specified file and revision.
FD.IQF	Outputs the first 10 columns of the FILE_DIRECTORY view, containing the name and other attributes of Vault files. The 10 columns are as follows:  Warning On a large data base, this command can take some time.
INREVIEW.IQF	Outputs the vote information about the specified file, file type, and revision that is currently being reviewed.
INREVWP.IQF	Outputs the vote information about a specified part and revision that is currently being reviewed.
LISTP.IQF	Outputs all the columns of the PROJECTS view, which contains names and descriptive information regarding all Vault Projects.
LISTU.IQF	Outputs the first 7 columns of the USERS view, containing names and other information about all Vault users, as follows:
LISTUP.IQF	Outputs a list of all Vault users on a specified project.
MARKA.IQF	Lists all files marked for archive.
MARKD.IQF	Lists all files marked for delete.
MESSAGES.IQF	Outputs the text of each message that was sent to the current user ID and logged in the message log.
REJREV.IQF	Lists all comments from the Reject_Review_Hist view.

**Table 6-1 IQF Query File Descriptions (Continued)**

IQF Query File	Description
SET . IQF	Information about the members of a specified file set: MEMBER_TYPE MEMBER_NAME MEMBER_REV MEMBER_DESC
SIGNOUT . IQF	Outputs all columns of the SIGNED_OUT_FILES view: LOCAL_FILE_NAME FILE_NAME FILE_REVISION USER_ID NODE_NAME ACTION_DATE ACTION_TIME
STAREV . IQF	All comments from the Status_Review_Hist view.
LOCATE . IQF	Lists the occurrence of a specified object on the various vaults in the Distributed Vault environment.
NOTIFYDL . IQF	Lists the notifications subscribed by the user that the user can delete.
NOTIFYQR . IQF	Lists all the notifications subscribed by the various users.

## Overview of IQFs

Locator query files are, essentially, stored Interactive Queries. Locator extends the Vault's stored query capability by providing facilities to specify title, column heading and arguments for use with the Vault Browser. These extensions are placed in special comments at the beginning of the query file. Each comment contains a keyword. Keywords must be preceded by `/*` and followed by a colon (`:`) and a space, and any keyword-specific information, followed by a space and a `*/`; all on a single text line.

### Example of an IQF Query

The following example retrieves the users of a specified project.

```
/* TITLE: EDM Users on a Specified Project */
/* ARG: Project ID */
/* HELPLIST: LISTP.HQF */
/* DEFAULT-PROJECTID: */
/* FIELD: user_id 12 User ID */
/* FIELD: user_name 25 Name */
select user_id,user_name from project_users - where
project_id='&1' - order by upper(user_id);
```

TITLE and ARG are Locator IQF keywords. TITLE, a general IQF keyword, applies to the entire query. ARG, an argument keyword, defines the label for the user-supplied argument.

# IQF Keywords

This section contains general keywords and argument keywords for the Query Facility with examples of their use:

**Table 6-2 General Keywords**

<b>Keyword</b>	<b>Description</b>
DATAONLY :	The presence of this keyword means that the output of an IQF file should be in the IQF "data only" format, that is, without titles or pagination. Only IQF files without a DATAONLY option use the IQF command. All other types automatically run in DATAONLY mode. DATAONLY mode connects directly to Oracle via the PDMDM account.
FIELD	This is required for each field in an SQF to parse the data received from the Vault. The syntax follows:  /* FIELD: field_name field_width field_caption */
TITLE :	The text to be displayed in the Query Selection window. Also displayed in the Argument window if there are any IQF arguments.
TYPE	This determines where to send the query. If the value of this keyword is DOD, then the query is sent to DOD. If this keyword is not there or does not have a value, then the query is sent to the local vault.

IQF arguments and their labels are defined by the keywords that follow. They can be used in IQF and SQF files.

**Table 6-3 Argument Keywords**

<b>Keyword</b>	<b>Description</b>
ARG	Defines an uppercase argument and specifies its label. Use ARG for uppercase database items like USERID and PROJECTID .
ARGL	Defines a mixed case argument and specifies its label. Use ARGL for mixed case database items like Description.

The following keywords apply to the arguments in Table 6-3, “Argument Keywords” on page 6-6:

**Table 6-4 Keywords for ARG and ARGL**

Keyword	Description
DEFAULT-PROJECTID:	This keyword, in the line after an ARG, sets the user text field to the currently specified Project ID.
DEFAULT-USERID:	This keyword in the line after an ARG sets the user text field to the currently signed on Vault user ID.
HELPLIST:	The HQF query executed when the drop-down button from the combo is selected. The following parameters can be passed to the query file: &projid, &selname, &selscope, &revision, &n.  At runtime, &projid is replaced with the Current Project-ID, &selname is replaced with the currently selected File or Part, &selscope is replaced with the currently selected file name, revision is replaced with the revision of the currently selected file or part, and &n is replaced with the value of the parameter number <i>n</i> from the Query Parameters window.
OPTION:	The specified values are appended to the drop-down list of a combo-box.

## Examples of IQF Keywords

```

/* TITLE: Project Files */
/* ARG: Project ID */
/* DEFAULT-PROJECTID: */
/* SET-VARIABLE: PROJECTID */
/* HELPLIST: LISTP.HQF */
/* OPTION: PROJ1 */
/* OPTION: PROJ2 */
/* FIELD: file_name 80 File Name */
/* FIELD: file_revision 20 File Revision */
/* FIELD: user_type 8 User Type */
set linesize 400
select file_name, file_revision, user_type from file_directory -
where class='PRO' and owner='&l' - order by
lower(file_name),file_revision

```

# Selection Query Files (SQFs)

Selection Query Files are used by Locator to select the view for the main window. You select a view in the Selection Attributes window.

## Types of SQFs

Standard Selection SQFs include three SQF classes for files (SF\* .SQF), parts (SP\* .SQF), and file sets (SS\* .SQF). Three SQFs come with Locator. You can also create your own. Standard Selection IQFs follow:

**Table 6-5 Standard Selection IQF Files**

<b>SQF File</b>	<b>Use</b>
SFA-PUB .SQF or SPA-PUB .SQF	When you select Public Files or Parts, respectively
SFB-PROJ .SQF or SPB-PROJ .SQF	When you select Project Files or Parts
SFC-PRIV .SQF or SPC-PRIV .SQF	When you select Private Files or Parts
SFD-SGN .SQF or SFD-SGN .SQF	When you select Files or Parts signed out to
SFE-LP .SQF or SPE-LP .SQF	When you select Files or Parts in Project Like
SFF-ED .SQF or SPF-ED .SQF	When you select Files or Parts in Entire Database Like (SQL)
SSS-FULL .SQF	When you select File Sets in Entire Database
SSS-SPC1 .SQF	When you select File Sets in Entire Database Like (SQL)
SFG-REG .SQF	Lists all the files registered from the local vault.
SPG-REG .SQF	Lists all the parts registered from the local vault
SFH-DVLT .SQF	Lists all the files in the Distributed Vault environment that a user can see
SPH-DVLT .SQF	Lists all the parts in the Distributed Vault environment that a user can see
SFI-LREV .SQF or SPI-LREV .SQF	Lists the Public Files with the latest Revision
SFJ-LREV .SQF or SPJ-LREV .SQF	Lists the Project Files with the latest Revision
SFK-LREV .SQF or SPK-LREV .SQF	Lists All the files with the latest Revision

## Writing SQF Files

Take care when modifying or adding SQF files. Changing the selection fields and their order changes not only the appearance of the main window, but it can interfere with Locator's ability to determine what you have selected.

The SF\* .SQF files, used for file selection, share the following basic query:

```
select file_name, file_revision, user_type, status_code, class,  
owner, part_number from file_directory
```

The SS\* .SQF files, used for file set selection, share the following basic SQL query:

```
select * from file_sets
```

The SP\* .SQF files, used for part selection, share the following basic SQL query:

```
select part_name, part_revision, user_type, status_code, class,  
owner, part_number from part_directory
```

The IQF keywords are applicable to SQF files.

## Internal Queries (RQFs and HQFs)

The following .RQF files are used by Locator:

**Table 6-6 RQF Files**

<b>RQFs</b>	<b>Use</b>
CPARTF .RQF	Fetches all member files of a specified part
ATTR .RQF and ATTRP .RQF	Gets attributes of a file or part, respectively
GETINFO .RQF and GETINFOP .RQF	Gets information on a file and part, respectively
GETLAR .RQF and GETLARP .RQF	Returns the latest revision of a specified file or part, respectively
PROJ .RQF	Retrieves the list of projects in the Vault
REJREV .RQF and REJREVP .RQF	Fetches review comments for a disapproved file or part respectively
REV .RQF	Retrieves the list of revision codes for the current project
SET .RQF	Fetches all members of a specified file set
STAREV .RQF and STAREVP .RQF	Fetches review comments for an approved file or part, respectively
STATUS .RQF	Retrieves the list of status codes for the current project
ULIST .RQF	Retrieves the list of user lists that exist in the Vault
USER .RQF	Retrieves the list of Vault users
BINDER .RQF	Lists all the binders
BINDMEMB .RQF	Lists the members of a given binder
NOPARENT .RQF	Lists the binders without a parent
EVENT .RQF	Lists the various Events

### Warning

You must not change the RQF files for any reason.

The following .HQF files are used internally by Locator and should not be modified. HQF queries should not use IQF keywords.

**Table 6-7 HQF Files**

<b>HQFs</b>	<b>Use</b>
LISTP.HQF	Used by LISTUP.IQF and SFB-PROJ.SQF to fetch a list of projects
STU.HQF	Used by SFC-PRIV.SQF and SFD-SGN.SQF to fetch a list of users
FILESET.HQFF	Used by the SET.IQF file to fetch a list of file sets
LISTULST.HQF	Used for finding out various user lists.



# Application Programmer Interface

---

This chapter describes the Locator application programmer interface (API). It includes the following information:

- Overview of the Application Programmer Interface (API)
- Differences Between Vault Programming and Locator API
- Locator API Logging Facility
- Custom Locator API
- Description of Locator API Calls
- Descriptions of Custom API Calls

# Overview of the Application Programmer Interface (API)

Locator provides an application programmer interface (API) for developers who want to write their own user interface applications on the PC.

The API is provided for Microsoft Windows applications. Runtime debugging is also available through an API log.

Locator API calls map directly to calls or capabilities provided by Vault Programming. To work with the API, developers should be familiar with Vault commands and Vault Programming.

The following manuals are recommended:

- *Vault Command Reference*
- *Vault Programmer Guide*

The Locator API uses the following dynamic-link library (DLL) files. These files are installed in the `WINDOWS` directory for MS-Windows or the `WINNT` directory for Windows NT:

- Locator API is the primary DLL for Windows 95 and Windows NT and is named `EDMAPI32.DLL`.
- The customization DLL named `EDMCUS32.DLL` supports all customization. A default version is installed when Locator is first installed.
- The transport DLL named `EDMXPT32.DLL` is called by the Locator API to communicate with Vault.

To customize Locator application programmer interface (API) calls, use Microsoft VC 4.0 or above. A new application can be written with any software package that supports DLL calls, for example, Microsoft C or Visual Basic.

# Differences Between Vault Programming and Locator API

In general, the Locator API calls map directly to functionality supported by Vault Programming, the differences between Vault Programming and Locator API are:

## Limitations for Command Keyword Options

Unless otherwise noted, the contents of the keywords correspond to those of the fields named in the *Vault Command Reference*. The keywords listed here have the following limitations in the Locator API.

Keyword	Limitation
output	Specified audit file output. Only F (file) or M (message) can be used.
append	Only R (replace) can be used.
selscope	For file transfers, only F (file), P (part), and S (file set) can be used.

## Locator API Parameters

The Locator API supports calls for getting and setting system parameters that apply to the Vault, Locator, and Vault Desktop Server.

The parameter facility is supported in two different forms:

- Directly with the `EDMgetopt` or `EDMsetopt` call
- With the `EDMciedm` call specifying the `Getoption` or `Setoption` command.

Sample calls are described in the section “Description of Locator API Calls” on page 7-8. The two most important parameters are set before signon: the Vault Host (`EDMOPTHost`) and the audit directory `EDMOPTauditdir`. The full list of parameter options is documented in the `EDMLIB.H` file included with the Locator/PC Programming SDK.

## Locator API Sessions

Locator API calls are session based. This allows an application to open and use more than one session at the same time. Sessions can all be to the same Vault host, or to different Vault hosts.

Multi-threaded Locator API calls must be used to set up multiple sessions. Multi-threaded sessions have a session ID. Multi-threaded API calls always begin with the prefix `mEDM` and require a session ID. Multi-threaded calls can also be used when there is only one session, though most developers prefer single-threaded calls for single sessions. Single-threaded API calls always begin with the prefix `EDM` and do not require a session ID.

The custom applications using Locator must do the following:

1. Start a session using either an `EDMinit()` call or the multi-threaded `mEDMinit()` call. A session ID unique to that session is returned by the `mEDMinit()`. The session ID must be used in all subsequent API calls for that session.

2. Before signing on, set two parameters: the Vault Host and the audit directory (for example, `C:\optegra\locator\audit`).

This can be done in one of the following two ways. Both methods have single-session and multi-threaded call forms. One method is with the `EDMsetopt()` or `mEDMsetopt()` call. The other method is with the `EDMciedm` or `mEDMciedm()` call specifying the `SETOPTION` command.

3. End the session just before your program exit using either an `EDMend()` call or the multi-threaded `mEDMend()` call.

## API Calling Interfaces

The Locator API provides five interfaces:

- `EDMcpdm()` and `mEDMcpdm()` that take the same control, command, and output status as Vault Programming
- `EDMfedm()` and `mEDMfedm()` that take the same parameters as the Vault Programming `Fedm()`
- `EDMciedm()` and `mEDMciedm()` that take command line user interface (CI) commands as parameters.
- `EDMsvedm()` and `mEDMsvedm()` that take the same parameters of an EDM command in the form of a variable argument list.
- Most Vault commands are also available in single-session and multi-threaded formats that permit each Vault command field to be put in a separate parameter string in the call for example, `EDMstore()` and `mEDMstore()`.

The Locator API does not support the `mEDMstore()` interface of Vault Programming.

## IQF Support

The Locator API provides access to the Vault Interactive Query Facility (IQF). The queries stored in a file can be executed against the vault's database and the results placed in an output file.

# Locator API Logging Facility

The Locator API contains a logging facility to assist the application developer with debugging. Each `metonymies` or `mEDMinit` call checks if logging is on. If logging is activated, a message box appears telling you the name of the log file. Log files are stored in the logging directory.

Please note: The log files must be deleted manually.

## Warning

Logs can reveal account and file password information.

The minimum level of logging consists of the following:

- Vault command names, `pi_retcode`, `pdm_retcode`, output message
- File transfer summaries: local file name, to or from, text or binary, byte size
- Delete file and test file
- Contents of all `setoption` and `getoption` values
- Size of all remote custom commands
- Contents of all fields to and from `EDMcmenu()`

Logging is controlled by the `EDMLIB_DEBUG` parameter in the `DEBUG` section of `WEDMLIB.INI` file located in windows home directory. The logging turns on if `EDMLIB_DEBUG` exists. Additional options can be added to the `EDMLIB_DEBUG` parameter as shown in the following table:

Option	Description
<code>-dir directory</code>	Directory on PC in which to put the log file (default is current working directory)
<code>-hexcmd</code>	Log vault commands, control, and output as hexadecimal dumps.
<code>-xpvt</code>	Log all calls to "transport" API library.
<code>-hexxpvt</code>	Hex dump of contents or all transport data.
<code>-hexremote</code>	Hex dump of contents of remote custom data.

For example:

```
[DEBUG]
EDMLIB_DEBUG= -dir c:\edmpc\log -xpvt
```

# Custom Locator API

Locator provides customization facilities through the custom API calls. The custom API is a DLL named `EDMCUS32.DLL`, which is called by the Locator API.

With the custom API, you can customize Locator in the following ways:

- Locator can be configured to add custom menu items to the File and custom menus. The code to implement these custom menu items must be placed in the custom API. Note that Locator also supports custom help for these custom items.
- Pre- and post-cpdm processing can be implemented in the custom API (for example, additional validations or operations).
- Mapping of Vault file names to PC file names can be customized.

The following items are needed:

- Locator/PC Programming SDK shipped with Locator/PC. This contains the source code and support files.
- Compilers, linkers, and software development kits (SDK) needed to build the custom API DLL.

There are two types of custom API calls: those that you want to modify for customization (`EDMccpdm`, `EDMclocalname`, and `EDMcmenu`) and others that support the custom call facility. You should not need to modify the latter.

To write your own custom API you should perform the following tasks:

1. Modify the source code.
2. Build the custom DLL - `EDMCUS32.DLL`.
3. Copy the DLL to the appropriate system directory, either `\WINDOWS` or `\WINNT`.

The custom API is never called by the actual application program. Applications always call the custom library through `EDMCUS32.DLL`, so as to handle any logging that can be active. Also, always use the multi-threaded (`mEDM`) form of the calls with the session ID (`SessID`) passed to the `EDMcinit()` routine.

## Description of Locator API Calls

The three types of API calls available in Locator are:

- Single-threaded (prefixed by EDM, for example, EDMaddfs)
- Multi-threaded (prefixed by mEDM, for example, mEDMaddfs)
- Custom API calls (prefixed by EDMc, for example, EDMcinit)

When working with single-threaded or multi-threaded API calls, use one type of call consistently, either single or multiple, but not both.

This section describes the single-threaded and multi-threaded API calls. The custom API calls are described in “Descriptions of Custom API Calls” on page 7-43.

Please note: In the following API calls the specified FAR\_PASCAL has been included. FAR\_PASCAL has the value FAR PASCAL.

### EDMaddfs( )

Adds the file set name to Vault. To add the members to the file set, use EDMaddmfs( ). Use EDMremfs to remove a file set from Vault.

#### Syntax

```
int FAR_PASCAL EDMaddfs(char *fsname, char *fsdesc, char *msg);  
  
int FAR_PASCAL mEDMaddfs(int SessID, char *fsname, char *fsdesc,  
char *msg);
```

#### Parameters Description

<b>fsname</b>	Unique file set name. Enter 80 or fewer alphanumeric characters.
<b>fsdesc</b>	Description of the file set. Enter 30 or fewer alphanumeric characters, including blanks.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMaddmfs( )

Adds a member to a file set. A member must be under the control of Vault and can be a file, CADDs part, file set, or user-defined entity.

### Syntax

```
int FAR_PASCAL EDMaddmfs(char *fsname, char *memtype, char
*memname, char *revision, char *memdesc, char *msg);
```

```
int FAR_PASCAL mEDMaddmfs(int SessID, char *fsname, char *memtype,
char *memname, char *revision, char *memdesc, char *msg);
```

### Parameters Description

<b>fsname</b>	Existing file set name. Enter 80 or fewer alphanumeric characters.
<b>memtype</b>	Code for the member type. Specify F (file) or P (part) or S (file set) or U (user-defined).
<b>memname</b>	Member Name is the name of the file, part, file set, or user-defined item to be added. Enter 80 or fewer alphanumeric characters.
<b>revision</b>	Revision code. If member type is F or P, enter 20 or fewer alphanumeric characters or leave blank for current revision.
<b>memdesc</b>	Description information. Enter 30 or fewer alphanumeric characters including blanks.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMchgfa( )

Changes the Vault file attributes. You can change the following file attributes: system type, user-defined type, description, part number, group technology code, and file type. This function can be used for files stored on the local node.

### Syntax

```
int FAR_PASCAL EDMchgfa(char *selscope, char *selname, char
*revision, char *filepw, char *output, char *append, char
*systype, char *usertype, char *filetype, char *filedesc, char
*partnum, char *gtcode, char *msg);
```

```
int FAR_PASCAL mEDMchgfa(int SessID, char *selscope, char
*selname, char *revision, char *filepw, char *output, char
*append, char *systype, char *usertype, char *filetype, char
*filedesc, char *partnum, char *gtcode, char *msg);
```

## Parameters Description

<b>selscope</b>	Selection Scope indicates the entity you are changing. Specify: F (file) or C (catalog) or P (part) or S (file set) or A (all of a project).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric characters.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric characters or leave blank for current revision.
<b>filepw</b>	File password. If file(s) has password, enter 8 or fewer alphanumeric characters.
<b>output</b>	Audit output information. Specify F (file) or M (message only).
<b>append</b>	Indicates whether to replace or append to the audit file, if audit destination (output parameter) is F. Currently only R (replace) is supported.
<b>systype</b>	System type code. Enter 8 or fewer alphanumeric characters. This parameter is optional.
<b>usertype</b>	User-defined type code. Enter 8 or fewer alphanumeric characters. This parameter is optional.
<b>filetype</b>	CGOS file type code. Enter 1 or 2 hexadecimal characters.
<b>filedesc</b>	File contents description. Enter 25 or fewer alphanumeric characters (including blanks). This parameter is optional.
<b>partnum</b>	Part description. Enter 20 or fewer alphanumeric characters (including blanks). This parameter is optional.
<b>gtcode</b>	Further file contents description or group technology code. Enter 32 or fewer alphanumeric characters (including blanks). This parameter is optional.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMchgfc1( )

Changes the file classification. This function can be used for files stored on the local node. This function can be used to change the classification of a file, part, catalog, project, or file set. To change the classification of a project file, you must have an access to the project files. To change the classification of a private file, you must be the owner of the file, and the current status code of the file must exist in the public authority scheme.

## Syntax

```
int FAR_PASCAL EDMchgfc1(char *selscope, char *selname, char
*revision, char *filepw, char *class, char *projid, char *output,
char *append, char *msg);
```

```
int FAR_PASCAL mEDMchgfc1(int SessID, char *selscope, char
*selname, char *revision, char *filepw, char *class, char *projid,
char *output, char *append, char *msg);
```

## Parameters Description

<b>selscope</b>	Indicates the entity you are changing. Specify F (file) or C (catalog) or P (part) or S (file set) or A (all of a project).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric characters.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric characters or leave blank for current revision.
<b>filepw</b>	File password. If file has password, enter 8 or fewer alphanumeric characters.
<b>class</b>	New file classification code. Specify: PUB (Public), PRO (Project), or PRI (Private).
<b>projid</b>	New project identifier. If new classification is PRO, enter 8 or fewer alphanumeric characters. If new classification is PUB or PRI, leave blank.
<b>output</b>	Audit output information. Specify F (file) or M (message only).
<b>append</b>	Indicates whether to replace or append to the audit file, if audit destination (output parameter) is F. Currently only R (replace) is supported.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMchgfpw( )

Changes the file password. This function can be used to change the password of a file, part, catalog, project, or file set. You must have write access to the files you are changing. To change the password of a project file, you must have access to the project files. To change the password of a private file, you must be the owner of the file. You cannot change the password of a file that is in use.

### Syntax

```
int FAR_PASCAL EDMchgfpw(char *selscope, char *selname, char  
*revision, char *filepw, char *nfilepw, char *output, char  
*append, char *msg);
```

```
int FAR_PASCAL mEDMchgfpw(int SessID, char *selscope, char  
*selname, char *revision, char *filepw, char *nfilepw, char  
*output, char *append, char *msg);
```

## Parameters Description

<b>selscope</b>	Indicates the entity you are changing. Specify F (file) or C (catalog) or P (part) or S (file set) or A (all of a project).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric characters.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric characters or leave blank for current revision.
<b>filepw</b>	File password. If file has password, enter 8 or fewer alphanumeric characters.
<b>nfilepw</b>	New file password. Enter 8 or fewer alphanumeric characters, or specify a sign (#) to remove password.
<b>output</b>	Audit output information. Specify F (file) or M (message only).
<b>append</b>	Indicates whether to replace or append to the audit file, if audit destination (output parameter) is F. Currently only R (replace) is supported.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMchgfreq( )

Changes the file revision. To use this function, a revision sequence should not be associated with the selection scope.

### Syntax

```
int FAR_PASCAL EDMchgfreq(char *selscope, char *selname, char  
*filepw, char *revision, char *newrev, char *msg);
```

```
int FAR_PASCAL mEDMchgfreq(int SessID, char *selscope, char  
*selname, char *filepw, char *revision, char *newrev, char *msg);
```

## Parameters Description

<b>selscope</b>	Indicates the entity you are changing. Specify F (file) or P (part).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric characters.
<b>filepw</b>	File password. If file(s) has password, enter 8 or fewer alphanumeric characters.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric characters or leave blank for current revision.
<b>newrev</b>	New revision code. Enter 20 or fewer alphanumeric characters or leave blank for no revision.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMchgfs()

Changes the file status code. The status code represents the phase of work within the authority scheme under which the file is stored. This function can be used to change the status code of a file, part, catalog, project, or file set.

### Syntax

```
int FAR_PASCAL EDMchgfs(char *selscope, char *selname, char  
*revision, char *filepw, char *currstat, char *newstat, char  
*comment, char *output, char *append, char *msg);
```

```
int FAR_PASCAL mEDMchgfs(int SessID, char *selscope, char  
*selname, char *revision, char *filepw, char *currstat, char  
*newstat, char *comment, char *output, char *append, char *msg);
```

## Parameters Description

<b>selscope</b>	Indicates the entity you are changing. Specify F (file) or C (catalog) or P (part) or S (file set) or A (all of a project).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric characters.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric characters or leave blank for current revision.
<b>filepw</b>	File password. If file(s) has password, enter 8 or fewer alphanumeric characters.
<b>currstat</b>	Current status code. Enter 8 or fewer alphanumeric characters unless you are changing a single file. Required if selection scope is C, A, or S.
<b>newstat</b>	New status code. Enter 8 or fewer alphanumeric characters.
<b>comment</b>	Status change description. Enter 240 or fewer alphanumeric characters (including blanks). This parameter is optional.
<b>output</b>	Audit output information. Specify F (file) or M (message only).
<b>append</b>	Indicates whether to replace or append to the audit file, if audit destination (output parameter) is F. Currently only R (replace) is supported.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMchgupw( )

Changes the current user's password. This function cannot change the signon password of other users.

### Syntax

```
int FAR_PASCAL EDMchgupw(char *userpw, char *nuserpw, char *msg);
```

```
int FAR_PASCAL mEDMchgupw(int SessID, char *userpw, char *nuserpw,  
char *msg);
```

## Parameters Description

<b>userpw</b>	Current Vault user password. Enter 8 or fewer alphanumeric characters.
<b>nuserpw</b>	New Vault user password. Enter 8 or fewer alphanumeric characters.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMciedm( )

Submits a Vault Programmatic Interface call using a string containing the equivalent to a CIEDM command line (command followed by `parameter=value` pairs). Single quotation marks can be used with `\'` escaping to include single quotation marks. Double quotation marks can be used with `\"` escaping to include double quotation marks.

## Syntax

```
int FAR_PASCAL EDMciedm(char *command, char *msg);

int FAR_PASCAL mEDMciedm(int SessID,
    char *command, char *msg);
```

## Parameters Description

<b>command</b>	Command to be submitted. Enter 8 alphanumeric characters.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

The usual Vault commands are recognized, for example:

```
char msg[300];
int i;
i = EDMciedm("CISIGNON USERID=SAM USERPW= FXB1", msg);
```

Two additional related Locator API commands are supported:

```
"SETOPTION optionname=value"
"GETOPTION optionname"
```

SETOPTION can be used to set options, while GETOPTION returns the value of the option. Options are described in the section “Locator API Parameters” on page 7-3.

## Example

```
i = EDMciedm("SETOPTION EDMOPTauditdir= C:\EDMPC\AUDIT",msg);
```

## EDMcopy( )

Copies a Vault file or part within the Vault. The file to be copied must not be currently in use. You must have Write access to the file you are copying. The format of the copy file name depends on the value of the selection scope:

- If the selection scope is F, the file copy will have the copy name.
- If the selection scope is P, each individual file within the part is copied to the copy name.
- If the selection scope is C, each individual file within the CGOS catalog is copied to the copy name with the original catalog name replaced by the new catalog name.

## Syntax

```
int FAR_PASCAL EDMcopy(char *selscope, char *selname, char  
*revision, char *filepw, char *copyname, char *copyrev, char  
*copycla, char *copyproj, char *copystat, char *copyfpw, char  
*output, char *append, char *msg);
```

```
int FAR_PASCAL mEDMcopy(int SessID, char *selscope, char *selname,  
char *revision, char *filepw, char *copyname, char *copyrev, char  
*copycla, char *copyproj, char *copystat, char *copyfpw, char  
*output, char *append, char *msg);
```

## Parameters Description

<b>selscope</b>	Indicates the entity you are copying. Specify F (file) or C (catalog) or P (part).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric characters.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric characters or leave blank for current revision.
<b>filepw</b>	File password. If file(s) has password, enter 8 or fewer alphanumeric characters.
<b>copyname</b>	Name of the file copy. Enter 80 or fewer alphanumeric characters.
<b>copyrev</b>	Revision code of copy. Enter 20 or fewer alphanumeric characters or leave blank for source file revision code.
<b>copycla</b>	File classification of copy. Specify PUB (public) or PRO (project) or PRI (private) or leave blank for source file classification.
<b>copyproj</b>	Existing project identifier of copy. If you are changing classification to PRO, enter 8 or fewer alphanumeric characters of an existing project identifier.
<b>copystat</b>	Status code. If you are changing, enter 8 or fewer alphanumeric characters. Leave blank for source file status code.
<b>copyfpw</b>	File password of copy. To give copy a password, enter 8 or fewer alphanumeric characters.
<b>output</b>	Audit output information. Specify F (file) or M (message only).
<b>append</b>	Indicates whether to replace or append to the audit file, if audit destination (output parameter) is F. Currently only R (replace) is supported.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMcpdm( )

Submits a Vault Programmatic Interface call directly.

### Syntax

```
int FAR_PASCAL EDMcpdm(struct pi_cntrl_struct *control, char  
*command, struct output_rec *output);
```

```
int FAR_PASCAL mEDMcpdm(int SessID, struct pi_cntrl_struct  
*control, char *command, struct output_rec *output);
```

## Parameters Description

<b>control</b>	It is a pointer to Programmatic Interface(PI) control structure. The structure is defined in <code>adpictl.h</code> .
<b>command</b>	The Vault command to be submitted in string format. Enter 8 alphanumeric characters.
<b>output</b>	Audit output information. Specify F (file) or M (message only). It is a pointer to the structure containing output information. The structure is defined in <code>pisample.h</code> .
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by <code>mEDMinit()</code> call. For single thread, specify 0.

## EDMend( )

Always terminate the Vault session with this call. This function closes the transport connection opened during the session and frees the memory storage that might have been used during the session.

### Syntax

```
int FAR_PASCAL EDMend();  
  
int FAR_PASCAL mEDMend(int SessID);
```

## EDMfedm( )

Submits a Vault Programmatic Interface call using an `argc/argv` vector list of `field=values`.

### Syntax

```
int FAR_PASCAL EDMfedm(char *command, int argc, char *argv, char *msg);  
  
int FAR_PASCAL mEDMfedm(int SessID, char *command, int argc, char *argv, char *msg);
```

## Parameters Description

- command** Command to be submitted. Enter 8 alphanumeric characters.
- argc** Number of Arguments passed to argv.
- argv** Argument vector list.
- msg** Returns message from Vault.
- SessID** Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMgetatlst()

Gets a file containing the attributes that would apply for the specified Classification Code, Project ID, Status Code, User Type, and Part Number.

### Syntax

```
int FAR_PASCAL EDMgetatlst(char *class, char *projid, char *statcd, char *usertype, char *partnum, char *attrfile, char *msg);
```

```
int FAR_PASCAL mEDMgetatlst(int SessID, char *class, char *projid, char *statcd, char *usertype, char *partnum, char *attrfile, char *msg);
```

## Parameters Description

- class** File classification code. Specify PUB (public), PRO (project), or PRI (private).
- projid** Existing project identifier. If classification is PRO, enter 8 or fewer alphanumeric characters.
- statcd** Status code. Enter 8 or fewer alphanumeric characters or leave blank for initial in-work code.
- usertype** User-defined type code. Enter 8 or fewer alphanumeric characters.
- partnum** Part description. Enter 20 or fewer alphanumeric characters (including blanks).
- attrfile** Name of a local file that contains attribute assignments. Enter 80 or fewer alphanumeric characters.
- msg** Returns message from Vault.
- SessID** Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMgetopt( )

Gets the value of the option. Returns the actual length of the option. Options are described in the section “Locator API Parameters” on page 7-3.

### Syntax

```
int FAR_PASCAL mEDMgetopt(int optno, char *option, int maximum);  
  
int FAR_PASCAL mEDMgetopt(int SessID, int optno, char *option, int  
maximum);
```

### Parameters Description

- optno** Option number. Twenty five options are available. Option numbers are declared in EDMlib.h file.
- option** Specify the value of the option to be set.
- maximum** Maximum number of bytes to be returned.
- msg** Returns message from Vault.
- SessID** Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMinit( )

A program must start with this call. If you use the multi-threaded EDMinit( ), it returns a nonzero SessID, which must be used in all subsequent calls.

### Syntax

```
int FAR_PASCAL EDMinit();  
  
int FAR_PASCAL mEDMinit();
```

## EDMiqf( )

Runs the IQF Query file using parameter arguments iqfargs with results to outfile. If dataonly is Y, the IQF is processed using the dataonly option.

### Syntax

```
int FAR_PASCAL EDMiqf(char *iqffile, char *outfile, char *iqfargs,  
char *dataflag, char *msg);  
  
int FAR_PASCAL mEDMiqf(int SessID, char *iqffile, char *outfile,  
char *iqfargs, char *dataflag, char *msg);
```

## Parameters Description

<b>iqffile</b>	Path of the IQF Query file. The maximum length is 128 characters.
<b>outfile</b>	Path of the output file that will have the result of the query. The maximum length is 128 characters.
<b>iqfargs</b>	Arguments for the query stored in IQF Query file. The maximum length is 255 characters.
<b>dataflag</b>	Data flag can be Y (yes - output file will have query data only) or N (no - output file will have column headers along with query data).
<b>msg</b>	Returns message from Vault
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0

## EDMlocalname( )

Application programmer interface to the custom library (EDMCUS32.DLL) routine EDMlocalname(). Generates local (PC) file name depending on the parameters specified.

### Syntax

```
int FAR_PASCAL EDMlocalname(char *selscope, char *selname, char  
*revision, char *usertype, char *end, char *overlay, char  
*lfdirname, char *pathname);
```

```
int FAR_PASCAL mEDMlocalname(int SessID, char *selscope, char  
*selname, char *revision, char *usertype, char *end, char  
*overlay, char *lfdirname, char *pathname);
```

## Parameters Description

<b>selscope</b>	Indicates the entity that you are creating. Specify F (file) or C (catalog) or P (part) or S (file set) or A (all of a project).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric characters.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric characters or leave blank for current revision.
<b>usertype</b>	User-defined type code. Enter 8 or fewer alphanumeric characters.
<b>end</b>	The end can be Y (yes) to indicate that this is the last mapping request of a set or N (no) to indicate there are more mappings to come. A set is often used for a file set when more than one Vault name in a file set maps to the same file name.
<b>overlay</b>	Overlay may be Y (may overwrite), N (may not overwrite), or U (unique). A unique value means that the PC will permute the filename (8-characters), as part of the file name until it gets a unique value.
<b>ldirname</b>	The local directory where the file is to be created.
<b>pathname</b>	If successful, sets the full path of the file.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMlog( )

Puts comments in an open log file. Returns 0 if comments were written in log file. Otherwise returns nonzero. For more information on logging, refer to “Locator API Logging Facility” on page 7-6.

## Syntax

```
int FAR_PASCAL EDMlog(char *string);
```

```
int FAR_PASCAL mEDMlog(int SessID, char *string);
```

## Parameters Description

<b>string</b>	The message to be put into log file.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMmarka( )

Marks a file for archiving. This function can be used to mark a file, part, catalog, project, or file set. When a file is marked for archiving, you can only have read access to it. A file cannot be marked for archiving if it is locked by the system or marked for deletion.

### Syntax

```
int FAR_PASCAL EDMmarka(char *selscope, char *selname, char
*revision, char *filepw, char *output, char *append, char *msg);
```

```
int FAR_PASCAL mEDMmarka(int SessID, char *selscope, char
*selname, char *revision, char *filepw, char *output, char
*append, char *msg);
```

### Parameters Description

<b>selscope</b>	Indicates the entity you are marking. Specify F (file) or C (catalog) or P (part) or S (file set) or A (all of a project).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric characters.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric characters or leave blank for current revision.
<b>filepw</b>	File password. If file has password, enter 8 or fewer alphanumeric characters.
<b>output</b>	Audit output information. Specify F (file) or M (message only).
<b>append</b>	Indicates whether to replace or append to the audit file, if audit destination (output parameter) is F. Currently only R (replace) is supported.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMmarkd( )

Marks a file for deletion. A file can be removed logically with the Delete command. A file cannot be marked for deletion if it is locked by the system or marked for archiving.

### Syntax

```
int FAR_PASCAL EDMmarkd(char *selscope, char *selname, char
*revision, char *filepw, char *output, char *append, char *msg);
```

```
int FAR_PASCAL mEDMmarkd(int SessID, char *selscope, char
*selname, char *revision, char *filepw, char *output, char
*append, char *msg);
```

## Parameters Description

<b>selscope</b>	Indicates the entity you are marking. Specify F (file) or C (catalog) or P (part) or S (file set) or A (all of a project).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric characters.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric characters or leave blank for current revision.
<b>filepw</b>	File password. If file has password, enter 8 or fewer alphanumeric characters.
<b>output</b>	Audit output information. Specify F (file) or M (message only).
<b>append</b>	Indicates whether to replace or append to the audit file, if audit destination (output parameter) is F. Currently only R (replace) is supported.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMmarkr( )

Marks a file to be restored. This function can be used to mark a file, part, catalog, project, or file set.

### Syntax

```
int FAR_PASCAL EDMmarkr(char *selscope, char *selname, char  
*revision, char *filepw, char *output, char *append, char *msg);
```

```
int FAR_PASCAL mEDMmarkr(int SessID, char *selscope, char  
*selname, char *revision, char *filepw, char *output, char  
*append, char *msg);
```

## Parameters Description

<b>selscope</b>	Indicates the entity you are marking. Specify F (file) or C (catalog) or P (part) or S (file set) or A (all of a project).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric characters.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric characters or leave blank for current revision.
<b>filepw</b>	File password. If file has password, enter 8 or fewer alphanumeric characters.
<b>output</b>	Audit output information. Specify F (file) or M (message only).
<b>append</b>	Indicates whether to replace or append to the audit file, if audit destination (output parameter) is F. Currently only R (replace) is supported.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMmenu( )

Called whenever a custom menu item is selected. **selscope**, **selname**, and **revision** are the currently selected items. **auditfile** is a string long enough to contain the path of an audit file if the custom menu code needs to return one.

## Syntax

```
int FAR_PASCAL EDMmenu(char *menuname, char *selscope, char
*selname, char *revision, char *usertype, char *statuscode, char
*auditfile, char *msg);
```

```
int FAR_PASCAL mEDMmenu(int SessID, char *menuname, char
*selscope, char *selname, char *revision, char *usertype, char
*statuscode, char *auditfile, char *msg);
```

## Parameters Description

<b>menuname</b>	Menu name is the one-word internal name of the menu item.
<b>selscope</b>	Indicates the entity you are selecting. Specify F (file) or C (catalog) or P (part) or S (file set) or A (all of a project).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric numbers.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric numbers or leave blank for current revision.
<b>usertype</b>	User-defined type code. Enter 8 or fewer alphanumeric characters.
<b>statuscode</b>	Status code. Enter 8 or fewer alphanumeric characters or leave blank for initial in-work code.
<b>auditfile</b>	Holds the path name of an Audit file.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMpurge( )

Logically deletes a file from Vault not previously marked for deletion. This command is basically a one-step deletion procedure, in contrast to the two-step procedure or the EDMmarkd( ) routine followed by the Delete command.

### Syntax

```
int FAR_PASCAL EDMpurge(char *selscope, char *selname, char  
*revision, char *filepw, char *output, char *append, char *msg);
```

```
int FAR_PASCAL mEDMpurge(int SessID, char *selscope, char  
*selname, char *revision, char *filepw, char *output, char  
*append, char *msg);
```

## Parameters Description

<b>selscope</b>	Indicates the entity you want to delete. Specify F (file) or C (catalog) or A (all of a project) or P (part) or U (user's private files).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric characters.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric characters or leave blank for current revision.
<b>filepw</b>	File password. If file(s) has password, enter 8 or fewer alphanumeric characters.
<b>output</b>	Audit output information. Specify F (file) or M (message only).
<b>append</b>	Indicates whether to replace or append to the audit file, if audit destination (output parameter) is F. Currently only R (replace) is supported.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMrcustom( )

Sends a remote custom command to the custom command code on the server. Sends the buffer `command` of length `len` and receives a `reply` up to length `maximum`. The return value is the actual length of the reply data.

## Syntax

```
int FAR_PASCAL EDMrcustom(char *command, int len, char *reply, int maximum);
```

```
int FAR_PASCAL mEDMrcustom(int SessID, char *command, int len, char *reply, int maximum);
```

## Parameters Description

<b>command</b>	Command to be sent. Enter 8 alphanumeric characters.
<b>len</b>	Length of the command that is passed.
<b>reply</b>	Reply for the command.
<b>maximum</b>	The maximum length of reply to be received.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMread( )

Used to read copied files from the Vault database to the local storage area. The file is not signed out. The data in the original file stored in Vault cannot be modified. You can read a file, catalog, part, or file set. You must have Read access to the file selection you want to read. The file must not be marked for deletion.

### Syntax

```
int FAR_PASCAL EDMread(char *selscope, char *selname, char
*lfname, char *ldirname, char *revision, char *filepw, char
*filetype, char *catlevel, char *datecrit, char *date, char
*output, char *append, char *overlay, char *cont, char *attrfile,
char *msg);
```

```
int FAR_PASCAL mEDMread(int SessID, char *selscope, char *selname,
char *lfname, char *ldirname, char *revision, char *filepw, char
*filetype, char *catlevel, char *datecrit, char *date, char
*output, char *append, char *overlay, char *cont, char *attrfile,
char *msg);
```

### Parameters Description

<b>selscope</b>	Indicates the entity you are reading. Specify F (file) or C (catalog) or P (part) or S (file set) or L (list).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric characters.
<b>lfname</b>	Name of the selection in your local storage area. Enter 80 or fewer alphanumeric characters.
<b>ldirname</b>	Local directory to which the file will be transferred. Enter 80 or fewer alphanumeric characters. Ignored if path is given in local file name for a single file.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric characters or leave blank for current revision.
<b>filepw</b>	File password. If file(s) has password, enter 8 or fewer alphanumeric characters.
<b>filetype</b>	File type code. If selection scope is C, enter 1 or 2 hexadecimal characters. Otherwise, all catalog files are copied.
<b>catlevel</b>	Number of catalog levels. If selection scope is C, enter 4 or fewer numbers. Otherwise, all catalog levels are copied.
<b>datecrit</b>	Indicates the time range for the selected catalog. Specify N (none specified) or B (before the date) or S (since the date). Optional if selection scope is C. Ignored if selection scope is F, P, or S.
<b>date</b>	USA or European standard date format. For USA format: month, followed by a slash (/), day, slash, year. For European format: day, followed by a period (.), month, period, year. If date scope is B or S and selection scope is C, enter the date.
<b>output</b>	Audit output information. Specify F (file) or M (message only).

- append** Indicates whether to replace or append to the audit file, if audit destination (output parameter) is F. Currently only R (replace) is supported.
- overlay** Indicates whether to overwrite any existing local files with the Vault file information. Specify Y (yes) or N (no).
- cont** Indicates whether to transfer an additional copy of a file already signed out to the user. Specify Y (yes) or N (no).
- attrfile** Name of a local file to which Vault can copy user-defined attributes. Enter 80 or fewer alphanumeric characters.
- msg** Returns message from Vault.
- SessID** Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMreadmsg( )

Used to read messages. You can read only messages sent to you.

### Syntax

```
int FAR_PASCAL EDMreadmsg(char *msgopt, char *msgnum, char  
*append, char *msg);
```

```
int FAR_PASCAL mEDMreadmsg(int SessID, char *msgopt, char *msgnum,  
char *append, char *msg);
```

### Parameters Description

- msgopt** Indicates what action to take with the message. Specify E (erase) or A (copy to audit file).
- msgnum** Message number. If MSGOPT is E, enter up to 10 numbers.
- append** Indicates whether to replace or append to the audit file, if audit destination (output parameter) is F. Currently only R (replace) is supported.
- msg** Returns message from Vault.
- SessID** Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMremfs( )

Removes a file set association from the Vault and deletes the file set name. The members of the set are not deleted but continue to exist in the database. You cannot remove a file set that is a member of another file set.

### Syntax

```
int FAR_PASCAL EDMremfs(char *fsname, char *msg);
```

```
int FAR_PASCAL mEDMremfs(int SessID, char *fsname, char *msg);
```

## Parameters Description

- fsname** Existing file set name. Enter 80 or fewer alphanumeric characters.
- msg** Returns message from Vault.
- SessID** Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMremmfs( )

Removes the member from the file set, but does not delete the member. The member still exists as a file or part on its own within Vault.

### Syntax

```
int FAR_PASCAL EDMremmfs(char *fsname, char *memtype, char *memname, char *revision, char *msg);
```

```
int FAR_PASCAL mEDMremmfs(int SessID, char *fsname, char *memtype, char *memname, char *revision, char *msg);
```

## Parameters Description

- fsname** Existing file set name. Enter 80 or fewer alphanumeric characters.
- memtype** Code for the member type. Specify F (file) or P (part) or S (file set) or U (user-defined).
- memname** Name of the file, part, file set, or user-defined item to be removed. Enter 80 or fewer alphanumeric characters.
- revision** Revision code. If member type is F or P, enter 20 or fewer alphanumeric characters or leave blank for current revision.
- msg** Returns message from Vault.
- SessID** Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMreplace\_dv( )

Transfers the updated version of a file from the local storage area to Vault, releases the signed-out file, and deletes the local copy.

### Syntax

```
int CALLING_CONVENTION EDMreplace_dv(char *selscope, char *selname, char *attrfile, char *append, char *output, char *vaultID, char *msg);
```

```
int CALLING_CONVENTION mEDMreplace_dv(int SessID, char *selscope,
char *selname, char *attrfile, char *append, char *output, char
*vaultID, char *msg);
```

## Parameters Description

<b>selscope</b>	Indicates the entity you are replacing. Specify F (file) or C (catalog) or P (part) or S (file set) or L (list).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric characters.
<b>attrfile</b>	Name of a local file to which Vault can copy user-defined attributes. Enter 80 or fewer alphanumeric characters.
<b>append</b>	Indicates whether to replace or append to the audit file, if audit destination (output parameter) is F. Currently only R (replace) is supported.
<b>output</b>	Audit output information. Specify F (file) or M (message only).
<b>vaultID</b>	Specify a remote Vault name in the distributed environment up to 32 alphanumeric characters, or specify "LOCAL" to gain access to the local vault.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMreqrvw( )

Requests a file review. It initiates the review cycle to advance the file to the next status level.

### Syntax

```
int FAR_PASCAL EDMreqrvw(char *selscope, char *selname, char
*revision, char *filepw, char *comment, char *output, char
*append, char *msg);
```

```
int FAR_PASCAL mEDMreqrvw(int SessID, char *selscope, char
*selname, char *revision, char *filepw, char *comment, char
*output, char *append, char *msg);
```

## Parameters Description

<b>selscope</b>	Indicates the entity to be reviewed. Specify F (file) or P (part) or S (file set).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric characters.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric characters or leave blank for current revision.
<b>filepw</b>	File password. If file has password, enter 8 or fewer alphanumeric characters.
<b>comment</b>	Review request description. Enter 240 or fewer alphanumeric characters (including blanks). This parameter is optional.
<b>output</b>	Audit output information. Specify F (file) or M (message only).
<b>append</b>	Indicates whether to replace or append to the audit file, if audit destination (output parameter) is F. Currently only R (replace) is supported.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMreserve( )

Reserves a unique file name in the Vault database to use in the future, and may also sign out the entry to your Vault user ID.

### Syntax

```
int FAR_PASCAL EDMreserve(char *pfname, char *revision, char
*class, char *projid, char *statcd, char *filepw, char* filetype,
char *systype, char *usertype, char *signout, char *filedesc, char
*partnum, char *gtcode, char *msg);
```

```
int FAR_PASCAL mEDMreserve(int SessID, char *pfname, char
*revision, char *class, char *projid, char *statcd, char *filepw,
char* filetype, char *systype, char *usertype, char *signout, char
*filedesc, char *partnum, char *gtcode, char *msg);
```

## Parameters Description

<b>pname</b>	The file should have an unique name in Vault. Enter 80 or fewer alphanumeric characters.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric characters or leave blank for current revision.
<b>class</b>	File classification code. Specify PUB (public), PRO (project), or PRI (private).
<b>projid</b>	Existing project identifier. If classification is PRO, enter 8 or fewer alphanumeric characters.
<b>statcd</b>	Status code. Enter 8 or fewer alphanumeric characters or leave blank for initial in-work code.
<b>filepw</b>	File password. If file has password, enter 8 or fewer alphanumeric characters.
<b>filetype</b>	File type code. Enter one or two hexadecimal characters.
<b>systype</b>	System type code. Enter 8 or fewer alphanumeric characters.
<b>usertype</b>	User-defined type code. Enter 8 or fewer alphanumeric characters.
<b>signout</b>	Designates whether the file should be signed out to the user reserving the file name. Specify Y (yes) or N (no).
<b>filedesc</b>	File contents description. Enter 25 or fewer alphanumeric characters (including blanks).
<b>partnum</b>	Part description. Enter 20 or fewer alphanumeric characters (including blanks).
<b>gtcode</b>	Further file contents description or group technology code. Enter 32 or fewer alphanumeric characters (including blanks).
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMreset( )

Resets the selected entity. This function is used when a user does not want a file signed out or does not want to change the Vault version with respect to changes made to local file version.

## Syntax

```
int FAR_PASCAL EDMreset(char *selscope, char *selname, char  
*output, char *append, char *msg);
```

```
int FAR_PASCAL mEDMreset(int SessID, char *selscope, char  
*selname, char *output, char *append, char *msg);
```

## Parameters Description

<b>selscope</b>	Indicates the entity you are resetting. Specify F (file) or C (catalog) or A (all of a project) or P (part) or S (file set).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric characters.
<b>output</b>	Audit output information. Specify F (file) or M (message only).
<b>append</b>	Indicates whether to replace or append to the audit file, if audit destination (output parameter) is F. Currently only R (replace) is supported.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMrsvp( )

Used to respond to a file review request on receiving a review message. You can reject or approve the file, and provide comments.

### Syntax

```
int FAR_PASCAL EDMrsvp(char *selscope, char *selname, char  
*revision, char *response, char *comment, char *msg);
```

```
int FAR_PASCAL mEDMrsvp(int SessID, char *selscope, char *selname,  
char *revision, char *response, char *comment, char *msg);
```

## Parameters Description

<b>selscope</b>	Indicates the entity you are reviewing. Specify F (file) or P (part) or S (file set).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric characters.
<b>revision</b>	Revision code. If selection scope is F or P, enter 20 or fewer alphanumeric characters. Ignored when selection scope is S.
<b>response</b>	Indicates whether you approve or reject the work. Specify A (approve) or R (reject).
<b>comment</b>	Review comments. Enter 240 or fewer alphanumeric characters (including blanks). This parameter is optional when response is A.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMsendmsg( )

Sends a message to a Vault user or members of a user list.

### Syntax

```
int FAR_PASCAL EDMsendmsg(char *userid, char *userlist, char  
*msgtext, char *msg);
```

```
int FAR_PASCAL mEDMsendmsg(int SessID, char *userid, char  
*userlist, char *msgtext, char *msg);
```

### Parameters Description

- userid** Existing Vault user identifier. Enter 12 or fewer alphanumeric characters.
- userlist** Existing user list name. Enter 24 or fewer alphanumeric characters.
- msgtext** Message text. Enter 240 or fewer alphanumeric characters (including blanks).
- msg** Returns message from Vault.
- SessID** Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMsetopt( )

Sets the option number to the value of `option`. Options are described in the section “Locator API Parameters” on page 7-3.

### Syntax

```
int FAR_PASCAL EDMsetopt(int optno, char *option);
```

```
int FAR_PASCAL mEDMsetopt(int SessID, int optno, char *option);
```

For example:

```
int i;  
i = EDMsetopt(EDMOPTauditdir, "C:\EDMPC\AUDIT");
```

## Parameters Description

<code>optno</code>	Option number. Twenty-five options are available. Option numbers are declared in <code>EDMlib.h</code> file.
<code>option</code>	Specify the value of the option to be set.
<code>msg</code>	Returns message from Vault.
<code>SessID</code>	Session ID returned by <code>mEDMinit()</code> call. For single thread, specify 0.

## EDMsignoff( )

Signs off from the vault, but does not disconnect from the host.

### Syntax

```
int FAR_PASCAL EDMsignoff(char *msg);  
  
int FAR_PASCAL mEDMsignoff(int SessID, char *msg);
```

## EDMsignon( )

Connects to the host (if not connected already) and signs on to the Vault domain as the specified user if the authentication is successful.

### Syntax

```
int FAR_PASCAL EDMsignon(char *host, char *domain, char *userid,  
char *password, char *msg);  
  
int FAR_PASCAL mEDMsignon(int SessID, char *host, char *domain,  
char *userid, char *password, char *msg);
```

## Parameters Description

<b>host</b>	Name of the host machine.
<b>domain</b>	Name of the domain to which you are signing on. Enter 32 or fewer alphanumeric characters or leave blank for the default Vault server.
<b>userid</b>	Unique Vault user identifier. Enter 12 or fewer alphanumeric characters.
<b>password</b>	User password. Enter 8 or fewer alphanumeric characters.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMsignout( )

Signs out files, without transferring the copy to the local storage area.

### Syntax

```
int FAR_PASCAL EDMsignout(char *selscope, char *selname, char  
*filepw, char *filetype, char *catlevel, char *datecrit, char  
*date, char *output, char *append, char *msg);
```

```
int FAR_PASCAL mEDMsignout(int SessID, char *selscope, char  
*selname, char *filepw, char *filetype, char *catlevel, char  
*datecrit, char *date, char *output, char *append, char *msg);
```

## Parameters Description

<b>selscope</b>	Indicates the entity you are signing out. Specify F (file) or C (catalog) or P (part) or S (file set).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric characters.
<b>filepw</b>	File password. If file(s) has password, enter 8 or fewer alphanumeric characters.
<b>filetype</b>	File type code. If selection scope is C, enter one or two hexadecimal characters. Otherwise, all catalog files are signed out.
<b>catlevel</b>	Number of catalog levels. If selection scope is C, enter 4 or fewer numbers. Otherwise, all catalog levels are signed out.
<b>datecrit</b>	Indicates the date range for the selected catalog. Specify N (none specified) or B (before the date) or S (since the date).
<b>date</b>	USA or European standard date format. For USA format: month, followed by a slash (/), day, slash, year. For European format: day, followed by a period (.), month, period, year. If date scope is B or S and selection scope is C, enter the date.
<b>output</b>	Audit output information. Specify F (file) or M (message only).
<b>append</b>	Indicates whether to replace or append to the audit file, if audit destination (output parameter) is F. Currently only R (replace) is supported.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMstore( )

Stores a file or part from the local storage area to the Vault database. To store a new version of a file that has already been stored in Vault and has been signed out for modification, use EDMupdate() or EDMreplace() function. A file cannot be stored if it exists with the same name in the archive table.

## Syntax

```
int FAR_PASCAL EDMstore(char *env, char *selscope, char *lfname,
char *signout, char *selname, char *pfname, char *revision, char
*class, char *projid, char *statcd, char *filepw, char *filetype,
char *attrfile, char *output, char *append, char *systype, char
*usertype, char *filedesc, char *partnum, char *gtcode, char
*msg);
```

```
int FAR_PASCAL mEDMstore(int SessID, char *env, char *selscope,
char *lfname, char *signout, char *selname, char *pfname, char
*revision, char *class, char *projid, char *statcd, char *filepw,
char *filetype, char *attrfile, char *output, char *append, char
*systype, char *usertype, char *filedesc, char *partnum, char
*gtcode, char *msg);
```

## Parameters Description

<b>env</b>	Code for application environment. Specify LOCAL or CADD5 or a site-defined value for custom parts.
<b>selscope</b>	Indicates the entity you are storing. Specify F (file) or C (catalog) or P (part) or D (directory) or L (list).
<b>lname</b>	Name of the file in your local storage area. Enter 80 or fewer alphanumeric characters.
<b>signout</b>	Indicates whether or not you want to keep the file signed out to you after you store it. Specify Y for yes or N for no.
<b>selname</b>	The selection name should be unique in Vault. Enter 80 or fewer alphanumeric characters or leave blank for local file name.
<b>pname</b>	Unique name of an existing part to which the single file being stored is added. Selection scope must be F and Sign out file must be N.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric characters or leave blank for first revision.
<b>class</b>	File classification code. Specify PUB (public) or PRO (project) or PRI (private).
<b>projid</b>	Existing project identifier. If classification is PRO, enter 8 or fewer alphanumeric characters.
<b>statcd</b>	Status code. Enter 8 or fewer alphanumeric characters or leave blank for initial in-work code.
<b>filepw</b>	File password. If file has password, enter 8 or fewer alphanumeric characters.
<b>filetype</b>	File type code. Enter one or two hexadecimal characters.
<b>attrfile</b>	Name of a local file that contains attribute assignments. Enter 80 or fewer alphanumeric characters.
<b>output</b>	Audit output information. Specify F (file) or M (message only).
<b>append</b>	Indicates whether to replace or append to the audit file, if audit destination (output parameter) is F. Currently only R (replace) is supported.
<b>systype</b>	System type code. Enter 8 or fewer alphanumeric characters. Optional.
<b>usertype</b>	User-defined type code. Enter 8 or fewer alphanumeric characters. Optional.
<b>filedesc</b>	File contents description. Enter 25 or fewer alphanumeric characters (including blanks). Optional.
<b>partnum</b>	Part description. Enter 20 or fewer alphanumeric characters (including blanks). Optional.
<b>gtcode</b>	File contents description or group technology code. Enter 32 or fewer alphanumeric characters (including blanks). Optional.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMsvedm( )

Submits a Vault Programmatic Interface call using a variable argument list consisting of the command followed by pairs of keyword and corresponding values separated by commas.

### Syntax

```
int FAR_CDECL EDMsvedm(char *command,...);
```

```
int FAR_CDECL mEDMsvedm(int SessID, char *command,...);
```

### Parameters Description

- command** Command to be submitted.
- msg** Returns message from Vault.
- SessID** Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMunmark( )

Unmarks the specified entity. This function unmarks the files that were marked with the EDMmarka(), EDMmarkd(), or EDMmarkr() functions. The UNMARK function must be carried out before the archive, delete, or restore commands.

### Syntax

```
int FAR_PASCAL EDMunmark(char *selscope, char *selname, char *revision, char *filepw, char *marktype, char *output, char *append, char *msg);
```

```
int FAR_PASCAL mEDMunmark(int SessID, char *selscope, char *selname, char *revision, char *filepw, char *marktype, char *output, char *append, char *msg);
```

## Parameters Description

<b>selscope</b>	Indicates the entity you are unmarking. Specify F (file) or C (catalog) or P (part) or S (file set) or A (all of a project).
<b>selname</b>	Selection name. If selection scope is not E, enter 80 or fewer alphanumeric characters.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric characters or leave blank for current revision.
<b>filepw</b>	File password. If file has password, enter 8 or fewer alphanumeric characters.
<b>marktype</b>	Indicates how selection was previously marked. Specify A (archive) or D (delete) or R (restore).
<b>output</b>	Audit output information. Specify F (file) or M (message only).
<b>append</b>	Indicates whether to replace or append to the audit file, if audit destination (output parameter) is F. Currently only R (replace) is supported.
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## EDMupdate()

Updates the specified entity. This function replaces the database version of the file with updated version, but the file remains signed out and a copy of it remains in the local storage area.

## Syntax

```
int FAR_PASCAL EDMupdate(char *selscope, char *selname, char  
*attrfile, char *append, char *output, char *msg);
```

```
int FAR_PASCAL mEDMupdate(int SessID, char *selscope, char  
*selname, char *attrfile, char *append, char *output, char *msg);
```

## Parameters Description

<b>selscope</b>	Indicates the entity you are updating. Specify F (file) or C (catalog) or P (part) or S (file set) or L (list).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric characters.
<b>attrfile</b>	Specify the name of a local file containing user defined attributes. Enter 80 or fewer alphanumeric characters.
<b>append</b>	Indicates whether to replace or append to the audit file, if audit destination (output parameter) is F. Currently only R (replace) is supported.
<b>output</b>	Audit output information. Specify F (file) or M (message only).
<b>msg</b>	Returns message from Vault.
<b>SessID</b>	Session ID returned by mEDMinit() call. For single thread, specify 0.

## Descriptions of Custom API Calls

The descriptions of the custom API calls for Locator follow.

### EDMccpdm( )

Called before and after each `cpdm( )` call. The `prepost` flag is 0 when called before `cpdm( )` and 1 when called after `cpdm( )`. Unlike Vault triggers, `EDMccpdm( )` can be used to modify the control, command, and output structure contents.

On a pre-`cpdm( )` call, setting a nonzero `control->pi_retcode` or a nonzero `control->pdm_retcode` causes the `cpdm( )` call to abort. Neither the `cpdm( )` nor the post-`cpdm( )` call is executed. Also, make sure you set an appropriate message in the output.

```
int FAR_PASCAL EDMccpdm(int CustID, int prepost, struct
pi_cntrl_struct *control, char *command, struct output_rec
*output);
```

### Parameters Description

<code>CustID</code>	Custom Session ID. This ID is returned from an <code>EDMccinit( )</code> call.
<code>prepost</code>	Prepost is called before and after each <code>cpdm( )</code> call. Prepost flag is 0 when called before <code>cpdm( )</code> and 1 when called after <code>cpdm( )</code> .
<code>pi_cntrl_struct</code>	Pointer to Programmatic Interface (PI) control structure. The structure is defined in <code>adpictl.h</code> .
<code>command</code>	The appropriate EDM command structure in string format.
<code>output</code>	Pointer to the output structure containing output information on <code>cpdm( )</code> call. The structure is defined in <code>pisample.h</code> .

### EDMccend( )

The Locator API software ends the custom session. This call frees the storage area used during your custom session:

```
int FAR_PASCAL EDMccend(int CustID);
```

## Parameters Description

**CustID** Custom Session ID. This ID is returned from an EDMcinit() call.

## EDMcfedm()

Called before and after each fedm() call. The Prepost flag is 0 when called before fedm() and 1 when called after fedm(). Unlike Vault triggers, EDMcfedm() can be used to modify the control, command, and output structure contents.

On a pre-fedm() call, setting a nonzero control->pi\_retcode or a nonzero control->pdm\_retcode causes the fedm() call to abort. Neither the fedm() nor the post-fedm() call is executed. Also, make sure you set an appropriate message in output.

```
int FAR_PASCAL EDMcfedm(int CustID, int prepost, struct  
pi_cntrl_struct *control, char *command, struct output_rec  
*output);
```

## Parameters Description

**CustID** Custom Session ID. This ID is returned from an EDMcinit() call.

**prepost** Prepost is called before and after each cpdm() call. Prepost flag is 0 when called before cpdm() and 1 when called after cpdm().

**pi\_cntrl\_struct** Pointer to Programmatic Interface (PI) control structure. The structure is defined in adpictl.h.

**command** The appropriate EDM command structure in string format.

**output** Pointer to the output structure containing output information on cpdm() call. The structure is defined in pisample.h.

## EDMccinit()

A custom session is initialized by the Locator API. It generates (and returns) a nonzero CustID that must be used in all subsequent calls.

```
int FAR_PASCAL EDMccinit(int SessID);
```

## Parameters Description

**SessID** Session ID. It is returned by EDMcinit() call. For single thread specify 0.

## EDMclocalname( )

Generates local (PC) file name depending on the parameters specified.

If successful, sets the full path in `pathname` and returns 0. On failure, sets an error message (using EDM “CPCCUSnnnE” notation) and returns -1.

## Syntax

```
int FAR_PASCAL EDMclocalname(int CustID, char *selscope, char
*selname, char *revision, char *usertype, char *end, char
*overlay, char *lfdirname, char *pathname);
```

## Parameters Description

<b>CustID</b>	Custom Session ID. This ID is returned from an EDMcinit() call.
<b>selscope</b>	The entity that you are creating. Specify F (file) or C (catalog) or P (part) or S (file set) or A (all of a project).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric numbers.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric characters or leave blank for current revision.
<b>usertype</b>	User-defined type code. Enter 8 or fewer alphanumeric characters.
<b>end</b>	The end can be Y (yes) to indicate that this is the last mapping request of a set; or N (no) to indicate there are more mappings to come. A set is most often used for a file set when multiple vault names in a file set map to the same file name.
<b>overlay</b>	Overlay can be Y (may overwrite), N (may not overwrite), or U (unique). A unique value means that the PC will permute the file name (8-character) part of the file name until it gets a unique value.
<b>lfdirname</b>	The local directory where the file is to be created.
<b>pathname</b>	If successful, sets the full path of the file.

## EDMcmenu( )

Called whenever a custom menu item is selected. `menuname` is the one-word internal name of the menu item (not the displayed text). `selscope`, `selname`, and `revision` are the currently selected item. `Auditfile` is a string long enough to contain the path of an audit file if the custom menu code needs to return one. `msg` contains the usual EDM-style message.

```
int FAR_PASCAL EDMcmenu(int CustID, char *menuname, char
*selscope, char *selname, char *revision, char *usertype, char
*statuscode, char *auditfile, char *msg);
```

## Parameters Description

<b>CustID</b>	Custom Session ID. This ID is returned from an EDMcinit() call.
<b>menuname</b>	Menu name is the one-word internal name of the menu item.
<b>selscope</b>	The entity you are selecting. Specify F (file) or C (catalog) or P (part) or S (file set) or A (all of a project).
<b>selname</b>	Selection name. Enter 80 or fewer alphanumeric numbers.
<b>revision</b>	Revision code. Enter 20 or fewer alphanumeric characters or leave blank for current revision.
<b>usertype</b>	User-defined type code. Enter 8 or fewer alphanumeric characters.
<b>statuscode</b>	Status code. Enter 8 or fewer alphanumeric characters or leave blank for initial in-work code.
<b>auditfile</b>	Holds the path name of an Audit file.
<b>msg</b>	Returns message.

## EDMcsetopt( )

Sets the `optno` option to the value of `option`. Options are described in the section “Locator API Parameters” on page 7-3.

```
int FAR_PASCAL EDMcsetopt(int CustID, int optno, char *option);
```

## Parameters Description

<b>CustID</b>	Custom Session ID. This ID is returned from an EDMcinit() call.
<b>optno</b>	Option number. Twenty-five options are available. Option numbers are declared in EDMlib.h file.
<b>option</b>	Specify the value of the option to be set.

# Working with Other Applications

---

This chapter provides information on the following topics for making Locator/PC work with other applications:

- Rulebases
- DDE Support

## Rulebases

A local rulebase is a collection of rules defined in a program. The name of the rulebase program is the value you assign to the Application Environment attribute (with the `store` command). A local rulebase handles all local file operations. All files are governed by a rulebase as defined in the Application Environment attribute (or part type).

Vault treats parts as collection of files and allows you to define custom parts. Parts are specific to a local application such as CADD5, MEDUSA, AutoCAD, Pro/ENGINEER, and so forth.

In Windows, the rulebase program is a `dll` that you call. All local rulebases in Windows are stored in the `%EPD_HOME%\lrb` subdirectory. Optegra Locator includes a standard, default local rulebase called `LOCAL.DLL` that is used with files. To customize a standard rulebase, or create your own rulebase, you need the Vault Client/PC Programming SDK separately. For details on rulebases including custom rulebases, refer to the *Vault System Administrator Guide*.

The Optegra Locator API calls the appropriate rulebase during file transfer commands via an entry point:

```
int _far _pascal LRB_entry(int SessionID, int argc, char  
**argv)
```

For further details, see the source code for the `LOCAL` rulebase on the EDMClient/PC Programming SDK.

In addition to the `dll` form of a client rulebase programs, Locator/PC API also supports an executable form of client rulebase programs.

For example, if you have a new executable of client rulebase program, `myrulebase.exe` for the application `MYRULEBASE`, add the following entry in `%WINDIR%\wedmlib.ini` in the `RULEBASE` section:

```
[RULEBASE]  
MYRULEBASE=c:%EPD_HOME%\lrb\myrulebase.exe
```

Locator first checks if the `exe` of the client rulebase is available (that is, `myrulebase.exe` is available) by checking `wedmlib.ini`. If `myrulebase.exe` cannot be found, Locator uses old client `dll` (that is, `myrulebase.dll`).

If you use the executable form of client rulebase programs, the same sources can be used for UNIX and PC clients. You can also use perl to write client rulebase programs. For example, if Application MYRULEBASE has a rulebase script myrulebase.pl, add the following entry in \$WINDIR\wedmlib.ini:

```
[RULEBASE]
MYRULEBASE=c:\perl5\bin\perl.exe c:\optegra\locator\lrb\
myrulebase.pl
```

For details on writing custom rulebases, refer to the *Vault System Administrator Guide*.

For each of the client rulebase programs (DLLs, executables, and perl scripts), there should be corresponding environment configuration(.ini) file in the lrb sub-directory of the Locator/PC installation directory. The structure of this file is:

```
[lrb]
description=Rulebase_Description
selscope=Valid_Vault_Selscope
store=Object_Browser_Application
store_prompt=Store_Prompt_to_Display
checkout=Checkout_Application
```

Contents of LOCAL.INI file of the default local rulebase - LOCAL.DLL, is listed below:

```
[lrb]
description=Local File
selscope=F
store=filebrws
store_prompt=Select a file to store
checkout=chkout-f
```

## DDE Support

Locator/PC supports Dynamic Data Exchange (DDE). DDE is a Windows facility. It allows two applications to communicate by exchanging data. A DDE session is called a conversation. The application that initiates the conversation is called the destination, while the application that responds is called the source. An application can be engaged in several conversations at the same time.

In order to have a conversation, both the applications must be running. When the destination begins a DDE conversation, it must specify two things:

- The DDE name of the application it wants to talk to
- The subject of the conversation ('topic')

Locator/PC can only take the role of source. Therefore the DDE facility has no impact on the Locator/PC user interface. The DDE name for Locator/PC is EDMLOG. Locator/PC supports multiple topics. Each topic requires a Locator/PC program instance. The topic, in effect, becomes the DDE session name. The default topic is SYSTEM. Locator/PC allows the execution of any EDM command using the command line syntax.

Data is exchanged in DDE via 'items'. Locator/PC supports two items:

- DDEConnectBroker
- DDECommandResponse

Note that all DDE names (i.e., applications, topics and items) are not case-sensitive. Every time you send a DDE message to Locator/PC, both items are set automatically. DDECommandResponse is the message returned by the EDM command.

Given below is a sample EDM command:

```
cistore LFNAME=c:\abc.txt SELNAME=sample.txt REVISION=1  
SELSCOPE=F CLASS=PUB STATCD=IW
```

For more information on the EDM command line syntax, consult the *Vault Command Reference*.

EDM commands must be sent in the context of an EDM session. Therefore you must first execute the following command:

```
CISIGNON USERID=yourEDMAccount USERPW=yourAccountPassword
```

Locator/PC also supports one additional command, `EXIT`. This command ends the DDE session and closes the Locator/PC application.

A sample DDE application (`ddeloc32.exe`) has been included in the distribution disk. This application is written in Visual Basic. The source files for the application have also been included.

All DDE sample files are located in the `\DDE` directory of Locator/PC Distribution Disk 1:

- `ddeloc32.exe` — the sample 32 bit application.
- `ddeloc.frm` — a Visual Basic source code file.
- `ddeloc32.mak` — the Visual Basic 4.0 (32 bit) make (support) file.

You can copy these files to a directory of your choice. The sample application starts Locator/PC, if necessary. In either case, the DDE conversation is controlling the associated Locator/PC instance. The sample program also uses your Locator/PC defaults for the audit directory.



# Administering Vault Desktop Server

---

This chapter provides the following information for administering the Vault Desktop Server:

- Overview of the Vault Desktop Server Configuration File
- Protocol Parameters
- Timer, Interval, and Limit Parameters
- Tracing Parameters

# Overview of the Vault Desktop Server Configuration File

The Vault Desktop Server software comes with parameters set to reasonable defaults to run the Server. System Administrators can customize the server environment with the parameters in the Vault Desktop Server configuration file on the Vault Server Host. The Vault Desktop Server configuration file, called `dtconfig`, is located in the path `$EDM_HOME/data/dtconfig` on a UNIX Vault and `%EDM_HOME%\data\dtconfig` on a Windows NT Vault.

## Desktop Support and Configuration

The Desktop Server (`dtserver`) provides communication to the Vault from Locator. It allows the execution of API commands and IQF queries. The Desktop Edmosrv (`dtedmosrv`) interfaces with the Vault to execute API queries and updates to the Oracle database.

### Server Locations

The servers start automatically when they have an AE entry in the `nsm.config` file. The server locations follow.

- UNIX

```
$EDM_HOME/bin/dtserver  
$EDM_HOME/bin/dtedmosrv
```

- Windows

```
%EDM_HOME%\bin\ntchild.exe  
%EDM_HOME%\bin\chldosrv.exe
```

To manually start and stop the servers, use the `nsmstart` and the `nsmstop` commands. For more information on these commands, see *Vault System Administrator Guide*.

### Log File Locations and Names

Vault Desktop Server keeps logs of all transactions between PC clients and the Vault. The log files are intended primarily to aid in diagnosing problems. You can configure the degree of verbosity of the log files.

The default locations for UNIX and Windows NT log files and names of log files are in the following table:

**Table 9-1 Default Locations and Names of Log Files**

	UNIX	Windows NT
Location of log files	/usr/tmp/dtserver_log	\temp\dtserver_log
Names of log files**	server.serverpid.sequence_number client.uniquenumber.clientpid.serverpid	server.serverpid.sequence_number client.uniquenumber.sequence_num.serverpid

More detailed logging can be enabled using the tracing parameters. These parameters are described in “Tracing Parameters” on page 9-8.

By default, log files are automatically deleted after seven days. If you want to keep any particular file longer than seven days, copy it to another directory.

## Configuration Basics for the Desktop Server

To configure the Desktop Server, set the values for the parameters in file \$EDM\_HOME/data/dtconfig. Descriptions on other types of parameters follow in this chapter. For information on customizing commands, see *Vault Command Reference*. The table that follows names the files related to the configuration of the Desktop Server.

**Table 9-2 \*Names of Files Related to Desktop Configuration**

File Name	Purpose and Comments
UNIX: \$EDM_HOME/data/dtconfig NT: %EDM_HOME%\data\dtconfig	The file in which you set the parameter values for the configuration.
UNIX: \$EDM_HOME/data/dtconfig NT: %EDM_HOME%\data\dtconfig	A file of information about configuration parameters.
UNIX: \$EDM_HOME/lib/libcvcustom.so NT: Not Applicable	A custom library for the Desktop Server. You can include your customized command processing here.
UNIX: /usr/tmp/clientPCname-pid NT: \temp\clientPCname-pid	Temporary files created during a session.

\*\*A TimeStamp generates the uniquenumber that is a part of the log file name.

uniquenumber = date +%m%d%H%M%S

## Categories of Configuration Parameters

Configuration parameters for the Vault Desktop Server fall into the following categories:

- Protocol parameters for the following:
  - To specify the TCP/IP port
  - To override the client's temporary directory on the Vault Desktop Server
  - To specify a “tickle” at the TCP/IP level for detecting sudden loss of the client
- Timing, interval, and limit parameters
- Tracing parameters for setting the level of detail for entries in the Optegra Locator log file

The configuration file contains lines of configuration parameters. Blank lines are ignored. Lines beginning with an asterisk (\*), a slash-asterisk (/\*), a semicolon (;), or a crosshatch (#) are treated as comment lines.

The following pages contain descriptions of the parameters. The tables below contain valid and default values for these parameters.

## Protocol Parameters

The protocol parameters are `TCP-port` and `TCP-keep-alives`, which specify the TCP/IP port and whether to have a “tickle” at the TCP/IP level for detecting sudden loss of the client.

**Table 9-3 Protocol Parameters**

Parameter	Default	Valid Values
<code>TCP-port</code>	2121	any port number
<code>TCP-keep-alives</code>	ON	ON/OFF
<code>Temp-directory</code>	<code>/usr/tmp</code> (for UNIX) <code>\temp</code> (for Windows NT)	path name

### `TCP-port` port number

This parameter overrides the default setting of the TCP/IP port. This information in the `services` file that is located in `/etc` directory under UNIX and in the `\winnt\system32\drivers\etc` directory under Windows NT. If the TCP/IP port is not set in this parameter or `services`, the default TCP/IP port is 2121.

### `TCP-keep-alives` ON|OFF

This parameter provides for a periodic “tickle” at the TCP protocol level to detect sudden loss of the client. The default setting is ON.

### `Temp-directory` temporary directory path

This parameter specifies the directory that is used for temporary storage for your PC on the Vault Desktop Server. The account that Vault and Vault Desktop Server are run under must have full access to this directory. This account is named `edm`. The default for this directory is `/usr/tmp` under UNIX and `\temp` for Windows NT.

## Timer, Interval, and Limit Parameters

The operation of the Vault Desktop Server involves a number of periodic activities as well as activities that occur when some time or size limit is reached. Each of these parameters is described below.

A number of internal timers and limits in the Vault Desktop Server can be configured. Time values can be set in seconds (default), minutes, hours, or days, for example:

30 seconds  
300 minutes  
5 days

Size values can be given in bytes (default) or kilobytes, for example:

100  
500KB

**Table 9-4** Timer, Interval, and Limit Parameters

Parameter	Default	Minimum	Maximum
Activity-timeout	3 minutes	20 seconds	8 hours
Inactivity-timeout	1 day	3 minutes	10 days
Configuration-check-interval	10 minutes	30 seconds	1 hour
Log-file-size	500KB	10KB	2000KB
Log-file-reset	1 day	1 hour	10 days
Log-file-retention	7 days	1 minute	1000 days

### Activity-timeout <sub>n</sub>

Each communications activity to the client is expected to complete before the expiration of this timer. If this timer expires, the client is presumed lost and the connection is shut down. The default value of this timer is 3 minutes.

### Inactivity-timeout <sub>n</sub>

When no activity is in progress, the server shuts down the connection to the client at the expiration of this timer. The default value is 1 day.

## Configuration-check-interval <sub>n</sub>

Controls how often Vault Desktop Server checks the parameters in the configuration file. If any parameters have been changed, the server process is dynamically updated. The default check interval is 10 minutes.

## Log-file-reset <sub>n</sub>

Controls how often Vault Desktop Server closes the current log file and opens a new one. If the limit of the log file size is reached before reaching this parameter limit, the current log file is closed, and a new one is opened. The default value for this parameter is 1 day.

## Log-file-size <sub>n</sub>

Controls the approximate maximum size that Vault Desktop Server permits the current log file to be. When the current log file is discovered to have exceeded this value, the Vault Server closes it and opens a new one. If the log file reset limit is reached before reaching this parameter limit, the current log file is closed, and a new log file is opened. The default size is 500KB.

## Log-file-retention <sub>n</sub>

Controls how long the log files are retained before the Vault deletes them. These files are located in the `/usr/tmp/dtserver_log` directory on a UNIX Vault and `\temp\dtserver_log` under Windows NT. The default is 7 days.

# Tracing Parameters

Vault Desktop Server logs all significant events in `server` and `client` log files. These files are located in the `/usr/tmp/dtserver_log` directory on a UNIX Vault and `\temp\dtserver_log` under Windows NT. The tracing parameters specify the level of detail for the logging of events in these files.

**Table 9-5 Tracing Parameters**

Parameter	Default	Values
Verbose	OFF	ON/OFF
Trace-commands	OFF	ON/OFF
Trace-recv-data	OFF	ON/OFF
Trace-send-data	OFF	ON/OFF
Trace-signed-out	OFF	ON/OFF
Trace-custom	OFF	ON/OFF

### Warning

Trace Logging can impact both disk storage and system performance. In general, use tracing parameters only to gather detailed diagnostic information for debugging purposes under the supervision of support staff.

## Verbose ON|OFF

Vault Desktop Server logs a brief description of each Vault command. Turning this parameter ON causes additional information to be logged. The default is OFF.

## Trace-commands ON|OFF

Vault Desktop Server logs a brief description of each Vault command that is executed. Turning this parameter ON causes a hexadecimal byte-by-byte dump of the contents of the input and output structures for each Vault command. The default is OFF.

## Trace-recv-data ON|OFF

Turning this parameter ON causes a hexadecimal byte-by-byte dump of all server data as it is received on the TCP/IP connection from the client PC. The default setting for this parameter is OFF.

## Trace-send-data ON|OFF

Turning this parameter ON causes a hexadecimal byte-by-byte dump of all server data as it is transmitted on the TCP/IP connection to the client. The default setting for this parameter is OFF.

## Trace-signed-out ON|OFF

For certain Vault commands, the server must query the RDBMS, and in the process, builds a temporary list of currently signed-out Vault entries. Turning this parameter ON causes the list to be printed to the client log file each time the list is built. The default is OFF.

## Trace-custom ON|OFF

Turning this parameter ON causes the hexadecimal logging of command and reply data for remote custom calls to Vault Desktop Server. The default is OFF.



# Messages and Troubleshooting

---

This appendix provides messages and troubleshooting advice for Locator/PC and includes the following:

- Overview of Messages
- Message Descriptions

## Overview of Messages

Vault returns a message in response to every Locator command. These messages are displayed in Locator in a message box. By default the error message are shown.

The error messages that follow are in ascending order by error number. Information and error messages are generated by both the Vault and Locator. Messages appear both in windows and in the Optegra Locator log. Messages from Locator always begin with the prefix CPC.

## Message Descriptions

The message descriptions that follow include information on possible causes of the problem as well as actions to remedy the situation.

CPCTCP001F Cannot find HOSTS entry for your PC.

Possible Cause: There are two possible causes of this error:

- Failure to include the IP address and name of your PC in your PC's HOSTS file
- Failure to include a new line (Enter character) after the name of your PC in the HOSTS file

User Action: Put an entry for the name of your PC in the HOSTS file. Include an explicit new line (Enter character) at the end of this entry, especially if this entry is the last one in the file.

CPCTCP002F Cannot find HOSTS entry for Vault Host.

Possible Cause: There are two possible causes of this error in your HOSTS file:

- No IP address and name of your Vault Host
- No new line (Enter character) after the name of your Vault Host

User Action: Put an entry for the name of your Vault Host in the HOSTS file. Make sure that there is an explicit new line (Enter character) at the end of this entry. It is especially important to include an explicit new line if this entry is the last one in the HOSTS file.

CPCTCP004F Connection rejected by Vault Host.

Possible Cause: The Vault Host is up and running, but the Locator Server is not running on the Host.

User Action: Contact your System Administrator.

CPCTCP005F Connection to Vault Host timed out.

Possible Cause: There is a network problem or the host is down, preventing Locator from reaching the Host.

User Action: Contact your System Administrator.

CPCTCP006F Connection to Vault Host failed, error = nnn.

Possible Cause: There is a TCP/IP error in your Vault network. The error number is reported by the TCP/IP network layer in the PC.

User Action: Refer to your TCP/IP documentation or contact your System Administrator.

CPCTCP007F TCP/IP driver not loaded.

Possible Cause: There are two possible causes:

- The TCP/IP driver has not been installed on your PC.
- A Locator DLL has been erased.

User Action: The respective remedies follow:

- Consult the documentation for your PC's network software. Verify that the TCP/IP driver is installed. Install if necessary.
- Reinstall Locator.

CPCTCP008F Not Connected to Vault Host. Please SIGN-ON.

Possible Cause: The system has stopped.

User Action: Contact your System Administrator.

CPCTCP009F Connection to Vault Host has been lost. Please SIGN-ON.

Possible Cause: The Vault Host system or the Locator Server has stopped.

User Action: Contact your System Administrator.

CPCTCP010F TCP/IP version on PC is nn. Need version xx.

Possible Cause: The TCP/IP software for Windows needs to be upgraded.

User Action: Contact your System Administrator. Consult the documentation for your PC's network software.

CPCIQF100E Cannot access temporary work file. Notify Vault Administrator.

Possible Cause: Disk space is inadequate. You have insufficient permission to access the work file.

User Action: Contact your System Administrator.

CPCIQF101E IQF file does not exist.

Possible Cause: The IQF file does not exist.

User Action: Contact your System Administrator.

CPCIQF102E No IQF file specified.

Possible Cause: An IQF file has not been selected to run.

User Action: Contact your System Administrator.

CPCIQF103E No IQF Output file specified.

Possible Cause: An IQF output file was not selected.

User Action: Refer to the Query Facility System information.

CPCIQF104E IQF Query Failed - see the audit file for further information.

Possible Cause: The IQF has failed.

User Action: Check the IQF.AUD audit file, refer to the Chapter “Query Facility System” or contact your System Administrator.

CPCIQF105E IQF Error: Could not copy the output file to your PC.

Possible Cause: There are two possible causes:

- The directory specified for this IQF is missing.
- Disk space on your PC is inadequate.

User Action: Change the output directory or make room on your disk. Then rerun the query.

CPCGET107E Processing not done - you do not have enough disk space available.

Possible Cause: Adequate disk space is not available on your PC to read or check out the files you requested.

User Action: Delete files from the specified drive, or choose a different drive with adequate disk space.

CPCSON109F Connection refused. Please correct your DATE and TIME.

Possible Cause: Your client node time is not within 24 hours either way of the Vault Server.

User Action: **Reset the time accordingly on either the client node or the Vault Server.**

CPCTCP110F Communications error in...

Possible Cause: **An unexpected network configuration error occurred as described in the text of the message.**

User Action: **Consult your network administrator.**

CPC125F Vault database connection failed.

Possible Cause: **Locator is unable to connect to the Vault RDBMS.**

User Action: **Consult your Vault administrator.**

---

# Glossary

---

## Administrative Server

The part of the Vault server that oversees administrative functions that do not impact files or parts, such as adding a user or creating a file set. See Vault server.

## 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 members have the same life cycles. 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.

## association

An independent relationship between a parent binder and the files and binders in that parent binder. All members can have the different life cycles. 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 item, that is, a file or part (group of files). A Vault item has in-built 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 Manager

The part of the Vault server that maintains the user-defined attribute database.

attribute set

A group of user-defined file attributes. Attribute sets are associated with files 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 with the 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 control access to files by users as part of an authority scheme. A user can access files with authority numbers less than or equal to their authority number. Authority numbers are assigned by a Vault 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. There is a public authority scheme for public files and a project authority scheme for each project. A project can use a job type for an authority scheme.

## binder

A user-defined collection of Vault files or other binders. 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.

## CADDs

Computervision Automated Design and Drafting System. A set of software packages for the CAE/CAD/CAM industry. Vault is particularly well-suited to managing CADDs parts.

## catalog

A logical group of CADDs files, parts, and other CADDs catalogs. Similar to a directory.

## check-in process

This operation returns file(s) to the Vault. Like an update, it creates a new version of the file, if the file has been modified. Unlike an Update, checking in a file in to the Vault unlocks the file for use by other Vault users. The Vault command name for check-in is REPLACE.

## check-out process

The retrieval of file(s) from the Vault for modification. Checking a file out of the Vault locks the file from modifications by other users. classification

## classification

Classification is a required, in-built, attribute of a Vault file or part. Each file and part in the Vault database is classified as public, private, or project.

## client

The Locator application, or the client, that requests a service from the Vault server.

## command list

A set of Vault commands with a command list name. Each Vault user is assigned one command list per authority scheme.

command trigger

A user-written application program for customizing the behavior of a Vault command. A command trigger can call an external process before or after execution of the command.

Data Distribution Server (DDS)

The Vault service that oversees transfers of files between the Vault and clients. Abbreviated as DDS. See Vault server.

Data Management Facility (DMF)

Part of the Data Distribution Service that determines if a transfer of a file can be performed. Abbreviated as DMF. See Vault server.

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 Vault, phases are grouped into a status level partition. The public authority scheme and each project has its own design cycle.

Distributed Object Directory (DOD)

A repository that contains locations of all registered files and parts. A Vault consults this directory to access items that you request.

Distributed Vault

Distributed Vault 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.

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.

By default, domain is the upper-case value of the host name. For example if your host name is kangaroo, then your domain name is KANGAROO.

**dormant**

The status designation for file that precedes the in-work status. Dormant files can be read but not checked out. Only the Project Manager and System Administrator can access dormant files. This optional status designation controls when the next revision can begin.

**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 Vault, a file has associated metadata, or attributes, such as its owner, history and status. These attributes protect and manage the file. Files can be grouped into parts and file sets.

**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. File sets allow you to execute a Vault command for all members of the file set with one execution of the command.

**host ID**

The computer node that runs Vault. Type `host id` to determine the identification number for your host.

**import services**

Import services import register files from the exporter Vault and replicate registered files and parts from the exporter Vault to other vaults (the importer) in the Distributed Vault environment.

**in-work**

The default initial status code and partition of a file in its design cycle. The next status code partition is Released. A file can optionally begin in the dormant stage. The life cycle moves from dormant, to in-work, to released.

**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 allows you to extract and output information from the Vault RDBMS about files, projects, authority schemes, and so forth.

interface

A set of rules to define and handle a group of files as a single part within an application environment such as CADD5. User-defined interfaces are implemented within Vault as an application or executable file.

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.

life cycle

See design cycle.

Locator

The Vault client software that allows a computer node to request data management services from a Vault server.

Log Services Manager

The Vault process that receives messages from each command.

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.

metadata

Information about a Vault file such as its classification, location, revision, status level, version, and so forth. Also known as *attributes*. 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 allows them to directly transfer information. Vault uses the TCP/IP network protocol.

## Network Services Manager (NSM)

NSM performs system and network management functions for nodes that access Vault. The Process manager is an example of an application under the control of NSM. See Vault server.

## node

A processing system attached to a network.

## notification

A report of the occurrence of an action or an event on a binder, such as change of a revision or the success or failure of a review. When you set up a subscription with Add Notification, you determine the receivers of the message and can set up an expiration date.

## notification list

A list of Vault user IDs that make up a user list. Users on the list are notified when a review is taking place 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.

## notification message

A message sent by Vault to inform you of the start of a review cycle or the occurrence of an action or event on a binder.

## Oracle

A relational database management system marketed by Oracle Corporation. Optegra applications use Oracle7 for basic database management functions.

## owner

The user ID to whom a file belongs. According to the classification scheme, public files have no owner, whereas private files are owned by a Vault user ID and project files are owned by a project ID.

part

A group of Vault files from within an application, for example, CADDs. Parts are controlled as a group and share the same name prefix. A file can belong to only one part.

Each part has a unique name and a classification, revision, status, and owner (public, project, or private) that it shares with its constituent files. Part types are defined with the environment attribute when a file is first stored in Vault and cannot be directly changed within Vault.

placeholder file

A reserved Vault file name. Use the Reserve Name option to hold a place for a future file. The reserved file name contains only metadata and no corresponding data file. Abbreviated in the database as P.

Private (PRI)

The classification assigned to files, parts, and directories that are owned by a single user ID. This classification, part of the authority scheme, includes read and write access by the user and the Vault administrator.

Project (PRO)

A file classification for a unit of work that includes its own files and parts, authority scheme, and assigned users. All files assigned to a project receive a project classification and a project ID and operate as a unit.

Public (PUB)

The file classification assigned to Vault files, parts, and directories that are accessible to every Vault user depending on their authority within the public authority scheme.

RDBMS

Relational Database Management System. Oracle and SQL/DS are RDBMS products used with Vault.

read authority

A level of permissions giving you the ability to read but not modify a file. Vault Read option copies a Vault file to your PC for viewing. This operation does not allow you to modify the file.

## registration

A process by which you register files in the Distributed Object Directory for access on Distributed Vaults. A registered item can be accessed by users in the Distributed Vault environment.

## relationship

The name of the connector between the parent binder, or container, and the items within it, such as files or other binders. Relationships can be an aggregation with a dependency on other items or an association independent of other items. See association and aggregation.

## released

A designation for a file or part is no longer being modified. This is the terminal status level partition. It follows the in-work stage.

## review list

A user list whose members are asked to review a portion of work. Each list member receives an action message from Vault.

## revision

Every Vault file follows a revision sequence based upon its authority scheme. The default for a new Vault file is the initial revision level in the sequence. When a released file is modified, it automatically moves to the next revision level in its revision sequence. A file can exist in the database with multiple revisions. Revision levels are represented by alphanumeric revision codes. Each revision of a file has a status level and a version.

## revision code

An alphanumeric designation indicating the number of times a file has been modified in a life cycle. Each time a file is released, it automatically receives the next revision level. A file can exist in the Vault with multiple revision codes.

## revision sequence

A series of revision codes in ascending order. One revision sequence applies in every authority scheme.

## server

The Vault software that provides data management services to the Locator clients. Each copy of Vault software makes up a domain. See domain.

SQL

See Structured Query Language (SQL).

status level

An 8-character code for the current phase of work on the Vault file or part within a design phases of In-work or Released. Each status code has an authority number. 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. Projects also provides file read and write authority levels by status level.

status level partition

A set of three design cycle phases to control whether you can modify a file: Dormant (optional), In-work (IW), and Released (R). Status partitions are further divided into status levels. The In-work and Released partitions must have at least one status level, whereas the optional Dormant stage has no status levels.

storage pool

The Vault term for the location of user files under its control. Vault 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.

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.

user ID

An account for an individual specific to a Vault host.

## 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 name of the computer node that runs Vault.

## Vault list

A list of vault domains in the same configuration for export and import of registered files within the Distributed Vault environment.

## Vault server

Vault server software and the database of user files and system data. Server processes handle the interaction between the Vault and Locator clients to provide data management services.

One or more Vault processes manipulate system data or user files according to your instructions. The Vault processes include the Administrative Server, Attribute Manager, Data Management Facility, Data Distribution Server, Network Services Manager, and Log Services Manager.

## version

A stage of file revision used in administrative tasks such as backup and recover. Vault uses a time stamp to identify the version. The Replace Item and Update Item options and other file recovery options create a new version, not readily visible to a user.

## write authority

A level of permissions giving you the ability to modify a file.



---

# Index

---

## A

- Archiving
  - restoring from archive 3-28
- Attributes 1-5, 1-6
  - editing 3-24
  - mark 3-27
  - user-defined
    - editing 3-20
- Audit files
  - directory 5-3
- Authority schemes 1-5

## B

- Binders
  - Change Class Attributes 4-9
  - Change Item Info 4-10
  - Create Query 4-12
  - Integration with other tools 4-14
  - Remove Item 4-8
  - Select User Defined Query 4-11
  - Update Display 4-10

## C

- CAD/CAM 1-4
- Check-out
  - released status level 3-19
  - undo 3-25
- Classification
  - setting 3-15, 3-23

- Command line interface
  - with Locator API 7-15
  - with Locator/PC API 7-4, 8-4

## D

- DDE 8-4
- Desktop Server
  - configuring 9-2
- Distributed Vault 1-3
  - Locate Object 1-3
  - Register Item 3-39
- Documentation, printing from Portable Document Format (PDF) file xvii
- Dynamic Data Exchange 8-4
- Dynamic-Link Libraries (DLL) 7-2
  - custom 7-7

## E

- EDMClient/PC
  - Programming SDK 7-7
- EDMWIN.INI file 5-2

## F

- File Browser 2-12
  - Edit menu 2-12
  - File menu 2-12
  - Font menu 2-12
- File menu

- Change Info 3-23
- Check-in 3-20
- Check-out 3-18
- Copy 3-17
- customizing 7-7, 7-45
- Mark 3-27
- Purge 3-26
- Read 3-21
- Request Review 3-34
- Reserve 3-16
- Reset 3-25
- Store 3-13
- Unmark 3-29
- Update 3-22
- File sets 1-4
  - check-in 3-20
  - marking 3-27
    - undoing 3-29
  - reviews 3-34
  - unmarking 3-29
  - updating 3-22
- Files 1-4, 1-6
  - archiving 3-28
  - attributes
    - editing 3-20, 3-24
  - check-in 3-20
  - copying 3-17
  - deleting 1-6, 3-26, 3-28
  - directory 5-3
    - configuring 2-15
  - names 3-14
    - mapping EDM to local 7-45
    - mapping vault to local 7-7
  - placeholder 3-16
  - project 1-5
  - public 1-5
  - purging 3-26
  - reserving name 3-16
    - undoing 3-16, 3-25
  - restoring from archive 3-28
  - reviews 3-34
  - storing
    - copy of existing Vault file 3-17
  - updating 3-22
- Follow-through application
  - check-in after check-out 3-19

## H

- HQF files 6-2, 6-11

## I

- Interactive Query Facility 6-2
- IQF 6-2, 7-5
  - command 6-2, 6-6
  - directory 5-3
  - files 6-2, 6-5
    - standard 6-3

## L

- Locator 1-2
  - API 7-1
    - custom 7-7, 7-43
  - Audit Files 2-12
  - configuring 2-16, 5-2
  - DDE support 8-4
  - dynamic data exchange 8-4
  - home drive 5-3
  - INI file 5-2
  - query facility 6-1
  - running 2-2
  - Signon 3-3
  - Vault messages
    - displaying 2-16
- Locator directories 5-3
- Locator menu
  - custom items 7-7
- Locator/PC
  - Programming SDK *xv*
    - DDE samples, 8-5
- Log files 3-9
  - directory 5-3
- Log menu 3-9
  - Save Log as 3-9
  - Show Log 3-9

## M

- Messages 3-31
  - in-box 3-30

- review 3-31
- RSVP 3-31
- sending 3-33
- Messages menu
  - Read Messages 3-30
  - Send Message 3-33

## N

- Network software
  - troubleshooting, A-1

## O

- Oracle
  - PDMDM account 6-2, 6-6

## P

- Parts 1-4
  - archiving 3-28
  - attributes
    - editing 3-20, 3-24
  - check-in 3-20
  - copying 3-17
  - deleting 3-26, 3-28
  - purging 3-26
  - restoring from archive 3-28
  - reviews 3-34
  - storing
    - copy of existing Vault file 3-17
  - updating 3-22
- Placeholder files 3-16
- Printing documentation from Portable Document Format (PDF) file xvii
- Project Administration
  - and classifications 1-5
- Project authority schemes 1-5
- Project classification 1-5
- Public authority scheme 1-5
- Public classification 1-5

## Q

- Query facility 6-1
  - files 6-2

## R

- Reserving file names 3-16
  - undoing 3-25
- Reviews 3-34
  - messages 3-31
  - voting 3-36
- Revisions
  - setting 3-23
- RQF files 6-2, 6-10
- Rulebases 8-2
  - defining custom parts for a local application 8-2
  - directory 5-3

## S

- Session log 3-9
  - saving 3-9
  - viewing 3-9
- Setup menu
  - Local File Access 2-13, 2-14, 3-21
  - Main Grid Columns 2-16
- Signon dialog
  - configuring 3-8
- SIGNOUT command 7-38
- SQF files 6-2, 6-8
- Status levels
  - released
    - and checking out 3-19
  - reviews 3-35
  - setting 3-15, 3-23

## T

- Tapes 3-28
  - archive 3-28

## U

### Unit of Measure

- Changing Attribute Values 2-11
  - dialog box 2-11
    - from Check In Item 3-20
    - from Store Item 3-15
    - from Update Item 3-22

Reading Attribute Values 2-11

Unit of Measure user attribute 2-10

### User Attributes

- Unit of Measure 2-10
  - Accessing Unit of Measure 3-15, 3-20, 3-22

Locator APIs 7-2

### Windows NT

- Locator APIs 7-2
- Vault installation 5-3

## V

### Vault

- accounts
  - password 3-7
- browser window
  - configuring 2-16
- commands 1-6, 3-9
  - customizing 7-43, 7-44
  - messages 2-16, 3-9
- messages
  - configuring 2-16
  - displaying 2-16
  - logging 3-9
- signing on 3-3

### Vault commands

- customizing 7-7

### Vault messages

- configuring 2-16

### Vault Programming

- and Locator API 7-2
- interfaces 7-4

### Views

- configuring 2-16
- customizing 6-8

### Visual Basic

- DDE sample application 8-5
- Locator applications 7-2

## W

### Windows