

Installing EPD[®].Connect,
EPD[®] Roles, and
EPD[®].Visualizer
Optegra[®] Release 6
DOC-60272-EN-060

Copyright © 2001 Parametric Technology Corporation. All Rights Reserved.

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

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

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

Registered Trademarks of Parametric Technology Corporation or a Subsidiary

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

Trademarks of Parametric Technology Corporation or a Subsidiary

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

Third-Party Trademarks

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

UNITED STATES GOVERNMENT RESTRICTED RIGHTS LEGEND

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

**Parametric Technology Corporation, 140 Kendrick Street, Needham, MA 02494-2714
8 January 2001**

Table of Contents

Preface

Related Documents	xvii
Book Conventions	xix
Online User Documentation	xx
Printing Documentation	xxi
Resources and Services	xxi
Documentation Comments	xxi

Introduction to EPD.Connect

Product Overview	1-2
Document Overview	1-2

Installing and Invoking EPD.Connect

Invoking EPD.Connect in the UNIX Environment	2-2
Changing Default Settings	2-2
Installing Other Applications	2-3
Invoking EPD.Connect in the Windows Environment	2-3
Changing Default Settings	2-3
Installing Other Applications	2-3
Using EPD.Connect	2-4

Contents

Installation Considerations	2-4
DATA_DIRECTORY Variable	2-4
Preinstallation Warning	2-4
Optegra Options	2-5
CAD Options	2-5
Multiple CAD Sessions	2-5
Configuring CAMU with Vault Client and EPD.Connect	2-5
Configuring CAMU with Vault Server and EPD.Connect	2-5
Using a Graphics Accelerator	2-6
EPD.Connect Web Browser	2-6
Setting Up the Web Browser	2-6
Multibyte Character Input Truncation	2-7
Controlling Timeouts	2-7
UNIX	2-7
Windows	2-8
Distributed Vault Environment	2-8
Installing EPD.Connect on Windows	2-8
Installing EPD.Connect and Vault on Windows NT	2-8
EPD.Connect on HP	2-9
Connecting to the Vault	2-9
For Windows	2-9
For UNIX	2-9
Displaying Desktop Server Messages	2-10
Considerations	2-10
Setting the Locale for a Language	2-11
UNIX	2-11
Windows	2-12
Additional Language-Specific Setting	2-12
Setting the Localized Language Variable	2-12
Remote Method Invocation (RMI) Servers for EPD.Connect	2-13
Configuring Vault File Object Types	2-13
PS and CAMU Object IDs	2-17

Configuring EPD.Connect

Overview	3-2
UNIX Initialization Files	3-2
Windows Initialization Files	3-3
Optegra Interface Files	3-4
UNIX	3-4
Windows	3-4
Establishing Local Defaults	3-5
Using the Command Line — UNIX	3-5
Using the Command Line — Windows	3-6
Using Initialization Files — UNIX	3-6
Using Initialization Files — Windows	3-6
Running Simultaneous Sessions	3-7
Understanding Supplied Files	3-7
cvepd.ini	3-8
UNIX	3-8
Windows	3-8
explorer.ini	3-9
epdconn.ini	3-9
cfg1024.ini	3-9
cfg1280.ini	3-9
cfg1152.ini	3-10
cfg800.ini	3-10
cfgmotif.ini—UNIX	3-10
cfgwin95.ini—Windows	3-10
cfgtemp.ini	3-10
Recognizing Precedence—UNIX	3-10
Recognizing Precedence—Windows	3-11

Configuring 3D Viewer, EPD.Visualizer, and CADDSS CAMU for EPD.Connect

Configuring the 3D Viewer _____	4-2
Enabling Colors on Windows NT Operating Systems _____	4-2
Generating and Locating Graphic Binary Files _____	4-2
Configuring for EPD.Visualizer _____	4-3
Setting the Variables _____	4-4
Turning Off Display Axes _____	4-4
Configuring for CADDSS CAMU _____	4-5
Relocating the Initialization File _____	4-5
Viewing Model Geometry _____	4-5
Setting CADDSS CVPATH _____	4-6
Setting CAMU DB DAEMON Host Name _____	4-6
Required Scripts for Vault and CADDSS _____	4-7
Installing EPD.Connect with CADDSS CAMU on the Same System as Vault _____	4-7
Order of Installation _____	4-8
Changing the DOD _____	4-8
Changing the Location of the Distributed Vault _____	4-8
Running navinstall _____	4-8
Installing EPD.Connect with CADDSS on a System with Remote Access to Vault _____	4-17
Installing EPD.Connect for the First Time _____	4-17
Upgrading an Installation _____	4-17
Accessing Vault from EPD.Connect _____	4-17
Listing Parts from within Vault _____	4-18
Sizing the CADDSS Windows _____	4-18
Sample Script _____	4-18
Considerations _____	4-21
Configuring the EPD.Connect Interface for CADDSS _____	4-21

Configuring EPD.Connect Interfaces

Configuring for CADD5 _____	5-2
Configuring the Vault Server _____	5-2
UNIX _____	5-2
Windows NT _____	5-4
Configuring the Vault Client _____	5-4
caddsr5.ini — Repository for CADD5 Interface _____	5-4
Using the Vault Command Line Interface _____	5-5
Configuring for MEDUSA _____	5-6
Configuring the Vault Server _____	5-6
UNIX _____	5-6
Windows NT _____	5-8
Configuring the Vault UNIX Client _____	5-8
medusarb.ini — Repository for the MEDUSA Interface _____	5-8
Customizing the MEDUSA Application Menus _____	5-10
Using the Vault Command Line Interface _____	5-12
Local Use of EPD Interfaces _____	5-12
Configuring for Pro/ENGINEER _____	5-12
Configuring the Vault Server _____	5-13
UNIX _____	5-13
Windows NT _____	5-15
Migration of Existing Interface Data _____	5-15
Configuring the Client _____	5-16
Extracting the Orientation Information from the Pro/ENGINEER Assembly _____	5-16
Guidelines for Installation _____	5-17
ptcarb.ini — Repository for the Pro/ENGINEER Interface _____	5-17
Using the Vault Command Line Interface _____	5-18
Local Use of EPD Interfaces _____	5-19
Configuring for CATIA _____	5-19
Configuring Vault Server _____	5-19
UNIX _____	5-19
Windows NT _____	5-21

Configuring the Vault Client _____	5-21
catiarb.ini — Repository for the CATIA Interface _____	5-21
Using the Vault Command Line Interface _____	5-22
Local Use of EPD Interfaces _____	5-22
Configuring for STEP AP203 _____	5-22
Configuring the Vault Server _____	5-23
UNIX _____	5-23
Windows NT _____	5-25
Configuring the Vault UNIX Client _____	5-25
stepb.ini — Repository for the STEP Interface _____	5-26
Configuring the STEP AP203 Translator _____	5-27
Local Use of EPD Interfaces _____	5-29
Shared Application Variable Settings _____	5-30
epdconn.ini _____	5-30
Repository Files for EPD.Connect Interfaces _____	5-33
cvepd.ini _____	5-33
epdconn.ini _____	5-34
Interface-Specific .ini Files _____	5-34
caddsrbi.ini _____	5-34
ptcarb.ini _____	5-34
medusarb.ini _____	5-35
catiarb.ini _____	5-35
steprbi.ini _____	5-35
Order of Precedence for Variables Used by the EPD.Connect Interfaces _____	5-35
Preprocess and Postprocess Variables for the EPD.Connect Interfaces _____	5-36

Configuring EPD.Connect for Vault

Setting the ANSPATH Variable _____	6-2
UNIX _____	6-2
Windows _____	6-2
Using CADD CAMU with Vault _____	6-2
Setting the Maximum Number of Users	

Within EPD.Connect _____	6-2
Multiple Vault Access: the pm.config File _____	6-3
Accessing Vault from a Different Domain _____	6-4
Running Two Vaults on the Same Server _____	6-4
Setting Up an Oracle/Vault Account _____	6-5
Launching and Starting Vault _____	6-5
Setting Up the Vault Interaction Panels _____	6-6
Setting Up the Vault Store Panel _____	6-6
Setting Up the Vault Transfers Panel _____	6-8
Setting Up the Vault Query Options Panel _____	6-8
Displaying the Labels Used to Describe Vault Items _____	6-10

Upgrading Navigator to EPD.Connect

Preparing to Upgrade _____	7-2
Exiting the Upgrade _____	7-2
Using the Navigator Refresh Tool _____	7-2
Starting the Upgrade _____	7-2
Accepting Defaults _____	7-3
Overriding Defaults _____	7-3
Entering Account and Path Information _____	7-3
Updating Oracle Tables _____	7-4
Checking for Navigator ADRAW Rulebase Support _____	7-4

Configuring EPD.Connect Application Defaults

Setting Colors _____	8-2
Specifying the Color Palette _____	8-2
Setting Colors for the Interface _____	8-2
Setting Component Text _____	8-3
Setting Fonts for the EPD.Connect Interface _____	8-3
Scaling the Windows _____	8-4

Contents

Specifying Directories _____	8-4
Primary Directories _____	8-4
Subdirectories _____	8-5
Setting a Mask for Files or Directories _____	8-6
Setting Database Defaults _____	8-7
Setting Up the Open Configuration Window _____	8-10
Controlling Component Selection _____	8-11
Controlling Geometric Accuracy _____	8-11
Controlling the Tessellation File Transfer _____	8-12
Controlling Clash Detection and Zoning _____	8-12
Setting Default Application Directories _____	8-14
Enabling Nested Reference Assemblies _____	8-16
Setting Information Browser Preferences _____	8-16

Customizing EPD.Connect Menus and Reports

Customizing the Toolbar _____	9-2
Default Toolbar File Locations and Pointers _____	9-2
The standard.def File _____	9-3
Creating a Customized Button Set _____	9-3
Customizing Menus _____	9-4
Editing the Contents of a Pulldown Menu _____	9-4
Scrolling Up and Down in a Menu List _____	9-5
Cancelling Menu Customization _____	9-5
Adding a Menu Option _____	9-5
Deleting a Menu Option _____	9-6
Editing a Menu Option _____	9-7
Sectioning a Menu _____	9-7
Creating a Custom Menu Command _____	9-8
Custom Menu Command Keywords _____	9-9
Example of a Custom Menu Command _____	9-10
Generating Tree Reports _____	9-10
Tree-Report Directory _____	9-11

Tree Report Command Format _____	9-11
Example of Report Generation _____	9-12
Generating Tree Comparison Reports _____	9-13
Tree Comparison Report Menu _____	9-13
Setting Colors to Reflect Comparison Results _____	9-14
Setting Comparison Tolerances _____	9-14
Displaying Reports _____	9-15
The Report Data File _____	9-15
Format _____	9-15
Header Block _____	9-16
Starting and Ending a Header Block _____	9-16
Defining the Assembly _____	9-16
Item Block _____	9-16
Starting and Ending an Item Block _____	9-17
Defining the Item _____	9-17
Example of an Item Block _____	9-17
Defining Characteristics of a Component _____	9-18
Defining the Part Associated with a Component _____	9-18
Defining Attributes for a Component _____	9-19
Defining Vault Attributes for a Component _____	9-20
Example of a Report Data File _____	9-20

Setting Event Triggers for EPD.Connect

About Event Triggers _____	10-2
Location of Event Triggers _____	10-2
Extract Triggers _____	10-3
Create Triggers _____	10-4
Editing Event Triggers _____	10-4

Installing EPD Roles

System Requirements _____	11-2
Software Configurations _____	11-2
Hardware Configurations _____	11-2

Contents

Disk and Swap Space	11-3
License Requirements	11-3
Oracle Runtime License Requirements	11-3
Installing EPD Roles	11-4
Procedure	11-4

Installing EPD.Visualizer

Setting Required Environment Variables	12-2
Installation Directory	12-2
Local Unaccelerated Graphics	12-2
Remote Workstation and Display	12-2
HP Workstation	12-2
Sun Workstation	12-3
Compaq Tru64 UNIX Workstation	12-3
IBM Workstation	12-3
SGI Workstation	12-4
Starting EPD.Visualizer	12-4
Setting Options at the Command Line	12-4
List Command Line Options	12-5
Specify a Language	12-5
Set Palette Position or Hide	12-5
Add Graphic or Meta Files	12-5
Add Files from a List	12-6
Set Starting Mode	12-6
Set Rendering Method	12-6
Set Buffering Method	12-7
Set Backface Culling	12-7
Display System Configuration	12-7
Add a License Path	12-7
Set Graphics Window Color	12-7
Special Workstation Considerations	12-9
HP Operating Environment	12-9
Memory Considerations	12-9
Graphics Acceleration	12-10

Compaq Tru64 UNIX Operating Environment _____	12-10
Screen Considerations _____	12-10
Movie Player _____	12-10
X Server _____	12-11
Sun Operating Environment _____	12-11
Graphics Library _____	12-11
Enable Retained Graphics _____	12-11
IBM Operating Environment _____	12-12
Retained Graphics _____	12-12
Default Fonts _____	12-12

EPD.Connect Product Structure File

About the Product Structure File _____	A-2
Format of the Product Structure File _____	A-2
Example of a Product Structure File _____	A-3
Converting Product Structure Trees to CAMU _____	A-4

EPD.Connect Action Codes

EPD.Connect Preconfiguration Checklist

Preface

Installing EPD.Connect, EPD Roles, and EPD.Visualizer provides the following:

- Instructions for configuring and customizing various EPD.Connect options (see chapters 1 to 10 and the appendixes)
- Instructions for installing EPD Roles (see Chapter 11)
- Instructions for installing EPD.Visualizer (see Chapter 12)

Basic installation of EPD.Connect and EPD Roles is described in *Installing Optegra Applications*. License management is described in *Using the License Manager*.

This book assumes that the machine-specific operating system has already been loaded and that you know how to use a file editor.

Related Documents

The following documents may be helpful as you use *Installing EPD.Connect, EPD Roles, and EPD.Visualizer*:

- *Installing Optegra Applications*
- *EPD.Connect User Guide*
- *Customizing EPD.Connect*
- *Optegra Release Notes*
- *Using the License Manager*

- *Installing CADD5 5i*
- *Managing CADD5 5i*
- *Concurrent Assembly Mock-Up User Guide and Menu Reference*
- *Parametric Modeling User Guide and Menu Reference*
- *Feature Based Modeling User Guide and Menu Reference*
- *NURBS User Guide and Menu Reference*
- *Explicit Modeling User Guide and Menu Reference*
- *Physical Properties User Guide and Menu Reference*
- *Rapid Prototyping User Guide and Menu Reference*
- *Design and Drafting User Guide and Menu Reference*
- *Imagedesign II User Guide and Menu Reference*

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> plot_config	Enter the text in a text box or on a command line. Where system output and user input are mixed, user input is 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 filename 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.

Convention	Example	Explanation
\$	<code>\$ rlogin remote_system_name -l root</code>	Indicates the Bourne shell prompt on command lines.
>	<code>> copy filename</code>	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`.
- Fill out and mail the PTC Documentation Survey located in the *PTC Customer Service Guide*.

Introduction to EPD.Connect

This chapter provides a brief introduction to EPD.Connect and the documentation describing how to use it. The following topics are presented:

- Product Overview
- Document Overview

Product Overview

EPD.Connect enhances productivity by connecting project team members with one another and to the information and applications relevant to their tasks. With EPD.Connect you can create and manipulate a graphical representation of a product (product structure). You can also see the model geometry associated with components of that structure. Optionally you can use additional Optegra and CAD products to further unify the project and product development process. EPD.Connect provides the following:

- A top bar of menu options, including an application-specific tool bar, for performing specific tasks and launching related applications
- An Information Browser for performing task-based and query functions
- A Product Structure window for viewing and manipulating product structures
- A 3D Viewer for displaying and visually manipulating associated 3D model geometry
- EPD.Visualizer for displaying and visually manipulating associated 3D model geometry and querying related attributes from databases.

Optional products that can be used with EPD.Connect include:

- CADDs CAMU
- CATIA
- EPD Interface to STEP
- MEDUSA
- Pro/ENGINEER
- Vault

Please note: For detailed information about EPD.Connect functionality and usage, see the *EPD.Connect User Guide*.

Document Overview

Use this manual after you have loaded EPD.Connect from the product distribution media and placed it on your local drive by means of an installation tool such as SLIC for UNIX and InstallShield for Windows. Use the manual to learn how to install and invoke EPD.Connect, and to configure it and the applications and interfaces that it may include.

Installing and Invoking EPD.Connect

This chapter contains information on invoking the default EPD.Connect installation. It also describes installation tasks that may be required after you have loaded the product. The following topics are presented:

- Invoking EPD.Connect in the UNIX Environment
- Invoking EPD.Connect in the Windows Environment
- Using EPD.Connect
- Installation Considerations
- Installing EPD.Connect on Windows
- Setting the Locale for a Language
- Remote Method Invocation (RMI) Servers for EPD.Connect
- Configuring Vault File Object Types

Please note: This chapter describes basic installation considerations. It assumes that you have already installed EPD.Connect and any optional packages from the product distribution media. The procedure for installing EPD.Connect is documented in *Installing Optegra Applications*.

Invoking EPD.Connect in the UNIX Environment

After you have loaded EPD.Connect, type the following command (or create an alias) to invoke EPD.Connect:

```
$EPD_HOME/scripts/Connect
```

The EPD.Connect splash screen appears, followed by the top bar menu and Information Browser window. The system is ready.

Please note: The required EPD_HOME variable is automatically set to `/opt/epd/dm/v60`. If you have loaded EPD.Connect into another location, set the environment variable to the software installation location before invoking EPD.Connect:

```
setenv EPD_HOME sw_location
```

Changing Default Settings

To override any EPD.Connect default settings, do one of the following:

- Create an empty file called `cvepd.ini` in your home directory and add the required variables to it. Obtain specific variables and values from the template files in `$EPD_HOME/data/reposit`. See “Installation Considerations” on page 2-4 and “Establishing Local Defaults” on page 3-5.
- Copy `$EPD_HOME/data/reposit/cvepd.ini` to your home directory and edit it to retain the desired variables and change their values.

To use the optional EPD.Visualizer, rather than the supplied 3D Viewer, you must also change default settings found in the template `explorer.ini` file. Place the required variables in your local `cvepd.ini` file. See “Configuring for EPD.Visualizer” on page 4-3.

Installing Other Applications

Refer to the following chapters of this book:

- Chapter 4, “Configuring 3D Viewer, EPD.Visualizer, and CADDSS CAMU for EPD.Connect”
- Chapter 6, “Configuring EPD.Connect for Vault”

Invoking EPD.Connect in the Windows Environment

After you have loaded EPD.Connect, invoke the application by choosing Start > Programs > EPD.Connect and Optegra > EPD.Connect.

The EPD.Connect splash screen appears, followed by the top bar menu and Information Browser window. The system is ready.

Changing Default Settings

To override any EPD.Connect default settings, edit your local `cvepd.ini` file (located in your Windows directory) to add more variables. Obtain specific variables and values from the template files in `%EPD_HOME%\data\reposit`. See “Installation Considerations” on page 2-4 and “Establishing Local Defaults” on page 3-5.

Installing Other Applications

Refer to the following chapters of this book:

- Chapter 4, “Configuring 3D Viewer, EPD.Visualizer, and CADDSS CAMU for EPD.Connect”
- Chapter 6, “Configuring EPD.Connect for Vault”

Using EPD.Connect

For detailed information about EPD.Connect functionality and usage see the *EPD.Connect User Guide*.

To learn about installing optional packages such as Vault and CAMU, and configuring for a browser, graphics card, and so on, see “Installation Considerations” on page 2-4.

To learn about initialization files and possible variable settings, see Chapter 3, “Configuring EPD.Connect.”

To learn about menu and report customization, see Chapter 9, “Customizing EPD.Connect Menus and Reports.”

Installation Considerations

You can use EPD.Connect with a variety of optional products (see “Product Overview” on page 1-2). In order for EPD.Connect to work correctly with these applications, you must configure it using various post-load installation procedures.

DATA_DIRECTORY Variable

If you have used CADD5 and set the `DATA_DIRECTORY` variable in your local `.caddsrc` files, be aware that the EPD.Connect startup script, `$EPD_HOME/scripts/Connect`, has been modified to disregard these settings. If you are setting the `DATA_DIRECTORY` variable intentionally, set it either only in the `EPDBase` file or on the command line when you start EPD.Connect.

The file `/usr/apl/cadd5/scripts/template.caddsrc` is the `caddsrc` template. It has been modified from CADD5 Rev 6.1.4, 7.1.0 and 8.0.0. It checks to see if `DATA_DIRECTORY` is set. If it is not, `.caddsrc` sets the original default value.

Preinstallation Warning

Please read the installation tasks and precautions carefully. If you do not follow them exactly as they are written, you could corrupt your database.

Optegra Options

You can optionally use the following applications from within the EPD.Connect environment:

- Vault—See “Configuring EPD.Connect for Vault” on page 6-1.

You can optionally use EPD.Visualizer, instead of the default 3D viewer, as the 3D viewing tool for EPD.Connect. See “Configuring the 3D Viewer” on page 4-2 and “Configuring for EPD.Visualizer” on page 4-3.

CAD Options

You can optionally use the following applications from within the EPD.Connect environment:

- CATIA—See “Configuring for CATIA” on page 5-19.
- CADDs CAMU—See “Configuring for CADDs CAMU” on page 4-5.
- MEDUSA—See “Configuring for MEDUSA” on page 5-6.
- Pro/ENGINEER—See “Configuring for Pro/ENGINEER” on page 5-12.

Multiple CAD Sessions

Do not launch multiple CAD sessions from within an EPD.Connect session.

Configuring CAMU with Vault Client and EPD.Connect

See “Required Scripts for Vault and CADDs” on page 4-7.

Configuring CAMU with Vault Server and EPD.Connect

See “Required Scripts for Vault and CADDs” on page 4-7.

Using a Graphics Accelerator

Use the `PVS_PICTURE` variable to specify your graphics card type. By default, the UNIX 3D Viewer is configured for X11 (nonaccelerated graphics on a UNIX machine). The Windows 3D Viewer is configured for msw (non-accelerated graphics on Windows) by default. The options are as follows:

Value	Platform
sbx	HP
sbx	HP
xgl	Sun
opengl	IBM
opengl	SGI
opengl	Compaq
msw	PC
opengl	PC

To take advantage of your hardware accelerator, set the following variable before application startup. Set the variable in your local `cvepd.ini` file, for example:

```
PVS_PICTURE=sbx
```

EPD.Connect Web Browser

Using the EPD.Connect Web Browser, you can gain access to the Vault through the World Wide Web or your intranet. You can view data residing in the Vault and download it to your client system.

Setting Up the Web Browser

Set the following variables in your local `cvepd.ini` file so that the Information Browser will communicate with your Web browser. See the example that follows for Netscape.

```
EPD_HTMLBROW_NAME=netscape
EPD_HTMLBROW_DIR=/usr/local/bin
EPD_HTMLBROW_CMDFMT=/usr/local/bin/netscape3.0.1/netscape
-noraise -remote 'openURL(%s)'
```

Please note: The command must include the absolute path name when specifying values for the EPD_HTMLBROW_CMDFMT variable.

Your Web browser must be running. Choose Tools > Web Browser so that your Information Browser can communicate with your Web browser.

Multibyte Character Input Truncation

On systems with multibyte character sets, the EPD.Connect GUI text-field sizes may allow more text entry than is actually allowed in a database table. EPD.Connect will truncate these fields to fit the table.

For example, in the Vault definition of user ID, you can use up to 12 characters, in terms of the Oracle table definition. At present, this means 12 bytes. With multibyte character sets, 12 bytes may represent a character-range count of 6 through 12, depending on the characters entered.

Controlling Timeouts

If EPD.Connect runs applications that handle large amounts of data, you may need to adjust the CMOM_CLIENT_TIMEOUT variable to prevent unwanted timeouts while data is exchanged. Set this variable before EPD.Connect is launched.

UNIX

Set the variable in one of these ways:

- In your environment, enter: `setenv CMOM_CLIENT_TIMEOUT xx`, where `xx` is the timeout value in seconds. The default value is 30.
- In your `cvepd.ini` file, add this line: `CMOM_CLIENT_TIMEOUT=xx`, where `xx` is the timeout value in seconds. The default value is 30.

Windows

Set the variable in one of these ways:

- Set this value through Control Panel > System > Environment.
- In the command shell, enter: `set CMOM_CLIENT_TIMEOUT=xx`, where `xx` is the timeout value in seconds. The default value is 30.

CMOM, the Common Message-Oriented Middleware, facilitates interprocess communication.

Distributed Vault Environment

In a distributed vault environment, you must list the DOD (Distributed Object Directory) as the first node in the `pm.config` file. Multiple DOD (Distributed Object Directory) functionality does not work. Only the first entry in the `pm.config` file is recognized.

Installing EPD.Connect on Windows

Find information and instruction for loading the Windows version of EPD.Connect in *Installing Optegra Applications*.

Installing EPD.Connect and Vault on Windows NT

If EPD.Connect and the Vault server for Windows NT are installed on the same machine, specify the complete path for all the rulebase executables referred in `$EPD_HOME/data/EDM.DEFAULTS`.

For example, if `$EPD_HOME` is `c:\epd\dm\v60` set `ENV(CADDS)` as follows:

```
ENV(CADDS)=C:\epd\dm\v60\bin\caddsif.exe
```

EPD.Connect on HP

When you install EPD.Connect on a HP, if you select a menu name from the menu bar and scroll through the menu commands while pressing the left mouse button, EPD.Connect crashes. To avoid this, deselect the selected menu command before you press the left mouse button to select another menu command.

Connecting to the Vault

The setup program enables Vault support. During setup you are asked for the name of the Vault to which to connect. You can change this setting later. Refer to “Setting the ANSPATH Variable” on page 6-2.

For Windows

To disable Vault support, set the following variables in the `cvepd.ini` file:

```
EDM_AVAILABLE=0  
CA_RB_ENABLED=N
```

To enable Vault support, set the following variables in `cvepd.ini`:

```
EDM_AVAILABLE=1  
CA_RB_ENABLED=Y
```

Please note: `EDM_AVAILABLE` is 0 by default.

For UNIX

To disable Vault support, set the following variables in your environment:

```
setenv EDM_AVAILABLE 0  
setenv CA_RB_ENABLED N
```

To enable Vault support, set the following variables in your environment:

```
setenv EDM_AVAILABLE 1  
setenv CA_RB_ENABLED Y
```

Please note: `EDM_AVAILABLE` is 0 by default.

Displaying Desktop Server Messages

In the Windows 95 environment, to display messages coming from the desktop server (JEUC charset) correctly in the Windows environment (SJIS charset), you must add the following to your desktop server configuration file:

```
servercharacteraset=JEUC
```

The Desktop Server configuration file is located at:

```
$EPD_HOME/data/dtconfig
```

The text file that describes how to set configuration options for the Desktop Server is:

```
$EPD_HOME/data/dtconfiginfo.txt
```

Considerations

- EPD.Connect cannot store files with names more than 8.3 DOS format.
- EPD.Connect is intended to run in a 32-bit Windows environment.
- The setup program does not configure non-English language settings. The default language is English C.
- To query Vault, `edmosrv` or `(dtedmosrv)` must be running on Vault.

Setting the Locale for a Language

You can set the locale for a language after installing EPD.Connect. The procedure for setting the locale depends on the operating system on which EPD.Connect is installed.

UNIX

To set the locale of a language on UNIX, perform the following:

1. To obtain a list of all available locales on a UNIX machine, use the command
2. If you want to set the locale for a language other than English, first unset the following variables:

```
LANG
LC_MONETARY
LC_TIME
LC_MESSAGES
LC_COLLATE
LC_CTYPE
LC_NUMERIC
LC_ALL
```

3. To set a particular locale for a language, on a Sun Solaris platform, set the parameters `LANG` and `LC_ALL` to one of the locales listed in the result of the previous command.
4. To verify that the locale has been set properly, use the following command:
locale
For example, the following settings set a French locale on a Sun workstation:
setenv LANG fr
setenv LC_ALL fr

Please note: If multiple locale choices exist for a particular language, use the one with `iso8859 - 1`, if available.

Please note: The value specified for `LANG` and `LC_ALL` must be the complete locale listed in the result of the `locale -a` command and not a part of it. For example, if the `locale -a` command lists `fr_FR.iso88591` as the only French locale, setting the `LANG` and `LC_ALL` to `fr` does not work.

Windows

To set the locale of a language on Windows, perform the following:

1. Set the default locale of your machine to the language in which you want to run EPD.Connect.
Settings>Control Panel>Regional Settings>Select language>Apply>OK
2. Select the language of your choice in the first screen of EPD.Connect PC Installation.

Additional Language-Specific Setting

The additional language-specific setting required for running EPD.Connect is independent of the operating system on which EPD.Connect is installed. It only depends on the language in which the EPD.Connect client is going to be invoked.

Setting the Localized Language Variable

To display the localized text in menus and messages, set the value of the localized language variable, `AWLANG`, in your `cvcpd.ini` file at the time of installation.

Set the value of the `AWLANG` variable as follows:

UNIX: `setenv AWLANG <language>`

Windows: `set AWLANG <language>`

where `<language>` is `C` for English, `french` for French, `german` for German, `japanese` for Japanese (UNIX), and `sjis` for Japanese (Windows). By default this variable is set to `C`, for English.

Please note: The above settings are case sensitive. For example, `AWLANG=sjis`

Remote Method Invocation (RMI) Servers for EPD.Connect

You can use Remote Method Invocation (RMI) servers to create distributed Java-to-Java applications. For details on configuring the RMI servers refer to the *Information Browser User Guide*.

Configuring Vault File Object Types

You can use the EPD.Connect Information Browser to associate types with file objects in Optegra Vault. This allows you to store and query distinct file object types in Optegra Vault and associate object type-specific behaviors to them. The file typing functionality requires the use of one of the five standard attributes:

- Description
- System Type
- User Type
- Part Number
- Class Code

Your site can choose to use any one of these five attributes for file typing through a preference. The standard configuration for Optegra 6.0 comes without file typing enabled. To enable file typing functionality for your site follow these steps:

1. Decide which of the five standard attributes to use for file typing information.

Note that the attribute choice limits the string size you can use for file typing. For example, the **Description** attribute enables you to use "Word Document" for MS word files, whereas the **System Type** attribute limits you to eight characters or less, such as "Word Doc".

2. Modify the corresponding object type files in the Registry, so that their ID field values match the file type names you have selected to use. (You can use the same type names that are already in the Registry and avoid having to make this change. For example, for a **PS** file object in the Vault, the Registry object `.ini, vault_PS.ini`, has `ID=PS`, so if you select to use **PS** as the file type name for PS files in the Vault, you need not modify the registry file.)
3. Re-create the Oracle `CONNECT_VIEW` synonym used by the Information Browser's Vault query service. The `CONNECT_VIEW`

synonym is configured for no file typing functionality; that is, it exists for the `Connect_Vault_NA` view. Depending on which standard attribute you expect to use, run the synonym configuration script `ibsynonym.pl` located under `$EDM_HOME/install` on the Vault server machine. You may run the script in two forms after changing to the `$EDM_HOME/install` directory:

a. With the command line option:

```
% ibsynonym.pl [option]
```

where [option] is an integer input, ranging from 1 through 6, or a two-character key, as listed here.

Integer	Key	Attribute	View
1	FD	Description	Connect_Vault_FD
2	UT	User Type	Connect_Vault_UT
3	ST	System Type	Connect_Vault_ST
4	PN	Part Number	Connect_Vault_PN
5	CD	Class Code	Connect_Vault_CD
6	NA	No File Typing	Connect_Vault_NA

b. Without the command line option:

The script displays the option list and prompts you to select one option.

```
% ibsynonym.pl
Java InfoBrowser Synonym creation module
0) Exit Installation
1) FD = FILE_DESC
2) UT = USER_TYPE
3) ST = SYSTEM_TYPE
4) PN = PART_NUMBER
5) CD = CLASS_CODE
6) NA = NOT APPLICABLE
Select your option now [6]:
```

When a number from 1 through 6 is selected, the script creates the synonym for the file typing attribute of choice or for no file typing. For example, suppose that a user decides to use **Description** as the file typing attribute.

Run the script:

```
% $EDM_HOME/install/ibsynonym.pl 1
```

or

```
% $EDM_HOME/install/ibsynonym.pl FD
```

or

```
% $EDM_HOME/install/ibsynonym.pl
Java InfoBrowser Synonym creation module
0) Exit Installation
1) FD = FILE_DESC
2) UT = USER_TYPE
3) ST = SYSTEM_TYPE
4) PN = PART_NUMBER
5) CD = CLASS_CODE
6) NA = NOT APPLICABLE
Select your option now [6]: 1
```

Please note: The preceding examples are specific to UNIX platforms. On Windows, execute the scripts as follows:

```
>%PERL_PATH% ibsynonym.pl [option]
```

or

```
>%PERL_PATH%-console ibsynonym.pl
```

where

PERL_PATH = The perl executable usually set to
\$EDM_HOME\perl5\bin\perl

4. Modify the value of the `FILE_TYPE_FIELD` preference in the `Vault.ini` file to point to the standard attribute you have chosen to use for file typing.

Attribute	Preference Value
Description	DESC
System Type	SYS_TYPE
User Type	USER_TYPE
Part Number	PART_NO
Class Code	CLASS_CD
<No file typing>	<null string> (as shipped)

5.

Please note: Because file typing is specific to your entire site (rather than individual users), control the `FILE_TYPE_FIELD` preference through the site preference file (for details, see “Enabling Nested Reference Assemblies” on page 8-16).

6. Decide on file type name strings (for example, "PSFile", "CAMU", "Word Doc") to be used as standards at your site and give this information to all users.
7. Write an SQL script to update file objects in your existing Vault. This script should set your selected standard attribute for a given type of file objects to the appropriate type name. For example, if you use the `Description` attribute and object type `PS` for all `PS` files in the Vault, the SQL script could be:

```
update DM_FILE_DIRECTORY set DM_FILE_DESC = 'PS' where,  
DM_FILE_NAME like '%$PS';
```

PS and CAMU Object IDs

The PS and CAMU object IDs in the EPD.Connect Registry in Optegra 3.0 have been set as PS and CAMU, respectively.

These object IDs can be customized, but where it is likely that both CAMU and PS files could exist in the same directory, the ID string for a PS object should include the "PS" substring and the ID string for a CAMU object should contain the "CAMU" substring.

For example, "PSFile" or "CAMUFile" or "ObjPS" or "CAMUdb" are valid, while "PFile" or "DBFile" are invalid and can cause incorrect results when the Vault STORE operation is performed on these objects from the Information Browser.

Configuring EPD.Connect

This chapter describes how to configure EPD.Connect using variables within your local `cvepd.ini` file. The following topics are presented:

- Overview
- Establishing Local Defaults
- Understanding Supplied Files

Overview

Configure EPD.Connect by setting variables in your local `cvepd.ini` file. Default settings are initially set in supplied (template) initialization files.

UNIX Initialization Files

The following supplied initialization files are placed in `$(EPD_HOME)/data/reposit` when you load EPD.Connect.

Initialization File	Description
<code>cvepd.ini</code>	Template for a user's local <code>cvepd.ini</code> file, to be used as a model.
<code>epdconn.ini</code>	Contains the majority of available EPD.Connect variables.
<code>explorer.ini</code>	Contains variables relative to a specific 3D Viewer.
<code>cfg1024.ini</code>	Contains variables relative to a 1024-resolution screen.
<code>cfg1280.ini</code>	Contains variables relative to a 1280-resolution screen.
<code>cfg1152.ini</code>	Contains variables relative to an 1152-resolution screen.
<code>cfg800.ini</code>	Contains variables relative to an 800-resolution screen.
<code>cfgmotif.ini</code>	Contains variables relative to a UNIX windows schema.

Please note: See “Recognizing Precedence—UNIX” on page 3-10 to learn how EPD.Connect recognizes variables within initialization files.

For infobrowser, look in `$EPD.HOME/cfg` for the supplied initialization files, including the following:

Initialization File	Description
<code>registry.ini</code>	Contains parameters for starting the registry server
<code>infobrowser.ini</code>	Contains startup parameters for the information browser
<code>vault.ini</code>	Contains startup parameters for the vault command server

Look for the registry initialization files in `$EPD.HOME/data/registry`.

Windows Initialization Files

The following supplied initialization files are placed in `$EPD_HOME/data/reposit` when you load EPD.Connect:

Initialization File	Description
<code>cvepd.ini</code>	Template for a user's local <code>cvepd.ini</code> file; on the Windows operating system, this file already exists in the Windows directory.
<code>epdconn.ini</code>	Contains the majority of available EPD.Connect variables
<code>explorer.ini</code>	Contains variables relative to a specific 3D Viewer.
<code>cfg1024.ini</code>	Contains variables relative to a 1024-resolution screen.
<code>cfg1280.ini</code>	Contains variables relative to a 1280-resolution screen.
<code>cfg1152.ini</code>	Contains variables relative to an 1152-resolution screen.
<code>cfg800.ini</code>	Contains variables relative to an 800-resolution screen.
<code>cfgwin95.ini—Windows</code>	Contains variables relative to a Windows environment windows schema.

Please note: See “Recognizing Precedence—UNIX” on page 3-10 to learn how EPD.Connect recognizes variables within initialization files.

For infobrowser, look in `$EPD.HOME/cfg` for the supplied initialization files, including the following.

Initialization File	Description
<code>registry.ini</code>	Contains parameters for starting the registry server.
<code>infobrowser.ini</code>	Contains startup parameters for the information browser.
<code>vault.ini</code>	Contains startup parameters for the vault command server.

Look for the registry initialization files in `$EPD.HOME/data/registry`.

Optegra Interface Files

The following `.ini` files are supplied for Optegra interfaces. These are installed only if the related interface is also installed.

Initialization File	Description
<code>caddsrp.ini</code>	Contains settings for the CADDs Interface.
<code>ptcarb.ini</code>	Contains settings for the Pro/E Interface.
<code>catiarb.ini</code>	Contains settings for the CATIA Interface.
<code>medusarb.ini</code>	Contains settings for the MEDUSA Interface.
<code>steprb.ini</code>	Contains settings for the STEP AP203 Interface.

UNIX

On UNIX platforms, these files are available in `$EPD_HOME/data/reposit`.

Windows

On Windows, these are available in the Windows System Directory.

Establishing Local Defaults

When you invoke EPD.Connect, the system searches for variables set in initialization files. You can use variables to establish settings for the application environment, such as the 3D viewing tool to use and whether to run two simultaneous EPD.Connect sessions. You can also set variables to determine basic interface characteristics, such as text font and size.

The key initialization file for EPD.Connect is your local `cvepd.ini`. Set your desired variables in your local `cvepd.ini` file.

See “Invoking EPD.Connect in the UNIX Environment” on page 2-2 and “Invoking EPD.Connect in the Windows Environment” on page 2-3 for more information.

Please note: To simplify the process of upgrading to future application revisions, do not edit any of the supplied initialization files. Any variables that you want to change should be set in your local `cvepd.ini` file.

Using the Command Line — UNIX

You can define an environment variable at the command line using the `setenv` command, as follows:

```
setenv variable_name value
```

For example, to configure a graphics accelerator, set the following environment variable as shown:

```
setenv PVS_PICTURE xgl
```

Although you can use environment variables to establish conditions, you typically set variables in an initialization file (namely, your local `cvepd.ini`) rather than at the command line.

Please note: Environment variables set at the command line (`setenv` for UNIX) before EPD.Connect is invoked override those set in initialization files. See “Recognizing Precedence—UNIX” on page 3-10.

Using the Command Line — Windows

You can define an environment variable at the command line using the `set` command, as follows:

```
set variable_name=value
```

For example, to configure a graphics accelerator, set the following environment variable as shown:

```
set PVS_PICTURE=xgl
```

Although you can use environment variables to establish conditions, you typically set variables in an initialization file (namely your local `cvepd.ini`) rather than at the command line.

Please note: Environment variables set at the command line (`set` for a Windows environment) before EPD.Connect is invoked override those set in initialization files. See “Recognizing Precedence—UNIX” on page 3-10.

Using Initialization Files — UNIX

You can define variables in your local `cvepd.ini`. Use template files in `$(EPD_HOME)/data/reposit/` to obtain desired variables and values.

When you invoke EPD.Connect from UNIX, it looks for the `.cvepd.ini` file in your home directory. If it exists, it is initialized and any commands therein are executed.

Please note: Your local `cvepd.ini` file may source other initialization files, such as `epdconn.ini` and `explorer.ini`. Supplied initialization files reside in `$(EPD_HOME)/data/reposit`. See “Recognizing Precedence—UNIX” on page 3-10.

Using Initialization Files — Windows

You can define variables in your local `cvepd.ini`. Use template files in `$(EPD_HOME)/data/reposit/` to obtain desired variables and values.

When you invoke EPD.Connect from Windows, it looks for the `cvepd.ini` file in your `C:\Windows` directory.

Please note: Your local `cvepd.ini` file may source other initialization files such as `epdconn.ini` and `explorer.ini`. Supplied initialization files reside in `$EPD_HOME/data/posit`. See “Recognizing Precedence—UNIX” on page 3-10.

Running Simultaneous Sessions

You can run two sessions of EPD.Connect on a single processor, if its resources permit. Set the following variables to do so:

Variable	Meaning
<code>EPD_TEMP</code>	Location of temporary files. Set it to a unique directory for each session.
<code>CMOM_DOMAIN</code>	Must be set to a unique session name for each session.

Warning

If you set `CA_TEMP` in the environment, you must unset it or set it to the value of `EPD_TEMP` if it is not already set to that value. It is recommended that you do not use `CA_TEMP`

In the following example, variables are set to run two sessions: `connect1` and `connect2` in two domains `EPDCONNECT1` and `EPDCONNECT2`:

```
CMOM_DOMAIN=EPDCONNECT1
EPD_TEMP=/usr/tmp/connect1
CMOM_DOMAIN=EPDCONNECT2
EPD_TEMP=/usr/tmp/connect2
```

Understanding Supplied Files

This section describes the objective of each supplied EPD.Connect initialization file.

Although several `.ini` files are supplied, all changes to the default settings should be made in your local `cvepd.ini` file only. For example, to change a

variable found in the template `epdconn.ini` file, add that variable (and set its desired value) in your local `cvepd.ini` file.

Please note: See “Recognizing Precedence—UNIX” on page 3-10 and “Recognizing Precedence—Windows” on page 3-11 for more information.

cvepd.ini

UNIX

Your local `cvepd.ini` file is the master file used to configure your EPD.Connect installation. You can customize it globally for a team or individually for each user. It contains several key variables. It can optionally call other initialization files.

You do not need a local `cvepd.ini` file. However, if you want to change default settings, create a `cvepd.ini` file in your home directory and place desired variables (and their values) in it.

You can optionally have a `cvepd.ini` file in your home directory that contains all default overrides. EPD.Connect only looks in your home directory for `cvepd.ini`. See “Changing Default Settings” on page 2-2.

Windows

The supplied `cvepd.ini` file is a template for the Windows user.

Your local `cvepd.ini` file is the master file used to configure your EPD.Connect installation. You can customize it globally for a team or individually for each user. It contains several key variables. It can optionally call other initialization files.

You must have a local `cvepd.ini` file in your `C:\Windows` directory to invoke EPD.Connect.

Please note: EPD.Connect does not look at `$EPD_HOME/data/reposit/cvepd.ini`. This is merely a template file for the Windows user.

Place a copy of the template `cvepd.ini` file in your Windows directory (`c:/win` or `c:/winnt`). Append any desired default overrides.

EPD.Connect only looks in your Windows directory for `cvepd.ini`. See “Invoking EPD.Connect in the Windows Environment” on page 2-3.

explorer.ini

Use the variables found in the template `explorer.ini` file to establish initialization defaults for the 3D Viewer supplied with EPD.Connect. Use its contents to also specify the 3D viewing tool – the default 3D Viewer or the optional EPD.Visualizer.

Put the specific variables that you want in your local `cvepd.ini` file.

You can also point to a 3D Viewer-specific initialization file using the `CVEPD_EXP_CONFIG` variable in your local `cvepd.ini` file. For example:

```
CVEPD_EXP_CONFIG=$HOME/explorer.ini
```

epdconn.ini

This file contains the majority of variables used to configure EPD.Connect. Put the specific variables that you want in your local `cvepd.ini` file.

This file is pointed to by the `CVEPD_MAIN_CONFIG` variable. For example:

```
CVEPD_MAIN_CONFIG=$CA_DIR/data/reposit/epdconn.ini
```

Please note: Variables within this file are described, by category, in Chapter 8, “Configuring EPD.Connect Application Defaults.”

cfg1024.ini

Contains screen resolution-specific variables.

cfg1280.ini

Contains screen resolution-specific variables.

cfg1152.ini

Contains screen resolution-specific variables.

cfg800.ini

Contains screen resolution-specific variables.

cfgmotif.ini—UNIX

Contains platform-specific variables, such as fonts.

cfgwin95.ini—Windows

Contains platform-specific variables, such as fonts.

cfgtemp.ini

Contains boilerplate variables for temp file names.

Recognizing Precedence—UNIX

If a variable is set in more than one place, its value is understood to be the first instance encountered. For example, if you set a variable in your local `cvepd.ini` file that is also set in the template `epdconn.ini` file, the value set in your local `cvepd.ini` is understood to be *the* value.

The order of precedence is as follows:

1. Environment variables set at the command line before invoking EPD.Connect.
2. Variables set in your local `cvepd.ini` file.

Please note: For a UNIX user, the file is either `cvepd.ini`, located in your home directory, or the file named by the `CVEPD_INIT` environment variable.

Please note: A local `cvepd.ini` file is optional for a UNIX user.

3. Variables set in the file pointed to by `CVEPD_MAIN_CONFIG`. This points to the template `epdconn.ini` file by default.
4. Variables set in the file pointed to by `CVEPD_WIN_CONFIG`. This points to the template `cfgmotif.ini` file.
5. Variables set in the file pointed to by `CVEPD_SCR_CONFIG`. This points to `cfg1280.ini` by default.
6. Variables set in the file pointed to by `CVEPD_EXP_CONFIG`. This points to the template `explorer.ini` file by default.

Recognizing Precedence—Windows

If a variable is set in more than one place, its value is understood to be the first instance encountered. For example, if you set a variable in your local `cvepd.ini` file that is also set in the template `epdconn.ini` file, the value set in your local `cvepd.ini` is understood to be *the* value.

The order or precedence is as follows:

1. Environment variables set at the command line before invoking EPD.Connect.
2. Variables set in your local `cvepd.ini` file.

Please note: For a Windows user, the file is `cvepd.ini`, located in your Windows directory.

Please note: A local `cvepd.ini` file is mandatory for a Windows user.

3. Variables set in the file pointed to by `CVEPD_MAIN_CONFIG`. This points to the template `epdconn.ini` file by default.
4. Variables set in the file pointed to by `CVEPD_WIN_CONFIG`. This points to the `cfgwin95.ini` file for Windows users.
5. Variables set in the file pointed to by `CVEPD_SCR_CONFIG`. This points to `cfg800.ini` by default.
6. Variables set in the file pointed to by `CVEPD_EXP_CONFIG`. This points to the template `explorer.ini` file by default.

Configuring 3D Viewer, EPD.Visualizer, and CADDSS CAMU for EPD.Connect

This chapter describes special configuration tasks. The following topics are presented:

- Configuring the 3D Viewer
- Configuring for EPD.Visualizer
- Configuring for CADDSS CAMU
- Configuring the EPD.Connect Interface for CADDSS

Configuring the 3D Viewer

The 3D Viewer (Product View window) is part of the basic EPD.Connect package, along with the Information Browser and the Product Structure windows. It enables you to view model geometry associated to components in the active Product Structure. It displays either `.gbf` files (graphic binary files) or `.gaf` (graphic ASCII files).

Changes that you make within the Product Structure window are reflected in the 3D Viewer by means of an update request.

Enabling Colors on Windows NT Operating Systems

On older Windows NT operating systems, you may have to change color settings to enable the 3D Viewer display.

1. In the Task Bar, choose Start > Settings > Control Panel > Display.
2. In the Display Properties sheet, choose the Settings tab.
3. In the Color Palette field, select High Color from the drop-down menu.

Generating and Locating Graphic Binary Files

You can generate graphic binary files from CADDs tessellation files (`_td` files) by specifying a conversion method in your local `cvepd.ini` file. By default, one is set for you in the template `explorer.ini` file.

Each model requires its own graphic binary file (or graphic ASCII file). If you are using CADDs, each model requires its own `_td` file for graphic binary (or ASCII) file generation.

Please note: The CADDShade II license (for generating `_td` tessellation files) is required and included with EPD.Connect. The CADDs-to-PVS translator (for generating `.gbf` and `.gaf` files from `_td` files) is required and included with CADDs.

The 3D Viewer uses variables found in the template `explorer.ini`. The applicable portion is shown next. Specify your desired conversion method within your local `cvepd.ini` file.

Please note: Method 3 is the default. It supports CAMU reference assemblies.

```
# Method 1 is gbf files in global coords in 1 directory
# (cgos assy)
# Method 2 is gbf files in local coords in 1 directory
# Method 3 is gbf files in local coords under the part
# (default)
# Method 4 is gbf files in global coords in 1 directory
# (unix name of assy)
#
# Note: For CADDs users in LOAD Method 3, Visualizer may
# be used with tessellation (_td) files. See below.
# CVDPA_PVS_TYPE={_pd.gbf,_td}
#
# CVDPA_LOAD_METHOD=1
# CVDPA_PVS_TYPE={.gbf,.gaf}
# CVDPA_CONVERT_TYPE=.gbf

# CVDPA_LOAD_METHOD=2
# CVDPA_PVS_TYPE={.gbf,.gaf}
# CVDPA_CONVERT_TYPE=.gbf

CVDPA_LOAD_METHOD=3
CVDPA_PVS_TYPE={_pd.gbf,_pd.gaf}
CVDPA_CONVERT_TYPE=_pd.gbf

# CVDPA_LOAD_METHOD=4
# CVDPA_PVS_TYPE={.gbf,.gaf}
# CVDPA_CONVERT_TYPE=.gbf
```

Please note: Graphic file locations are dependent on the `CVDPA_LOAD_METHOD` you use.

Configuring for EPD.Visualizer

The EPD.Visualizer is a more powerful graphics viewing tool than the basic 3D Viewer supplied with EPD.Connect. (“Configuring the 3D Viewer” on page 4-2 discusses configuring the 3D Viewer.)

If you purchase and install EPD.Visualizer, you can also perform advanced clash detection and zoning operations, cross sectioning, and superior animation operations.

EPD.Visualizer uses variables found in the template of the `explorer.ini` files. See “explorer.ini” on page 3-9.

Setting the Variables

Set the `CVDPA_VIS_TARGET` and `CVDPA_EPD3dViewer_COMM` variables in your default `cvepd.ini` file as follows:

```
CVDPA_VIS_TARGET=Visualizer
CVDPA_EPD3dViewer_COMM=$CA_SCRIPTS/Visualizer &
```

An excerpt from the template `explorer.ini` file follows:

```
# set the name of the viewer to send message to
#
#CVDPA_VIS_TARGET=EPD3dViewer
CVDPA_VIS_TARGET=Visualizer
#
# set name of the visualizer executable and directory to
# start in
#
#CVDPA_EPD3dViewer_COMM=$CA_BINDIR/epd3dviewer
$CA_DATA/resource/viewmain.r&
CVDPA_EPD3dViewer_COMM=$CA_SCRIPTS/Visualizer &
```

Please note: By default, the variables are set in the file pointed to as `CVEPD_EXP_CONFIG=$EPD_HOME/data/reposit/explorer.ini`

Turning Off Display Axes

If you are using EPD.Visualizer, the `PVS_AXIS_DISPLAY` variable enables you to turn off the display of the three axes that appear at the bottom left of the window.

```
PVS_AXIS_DISPLAY=on
```

Configuring for CADDs CAMU

If you purchase and install CADDs CAMU with EPD.Connect, you can display and manipulate CAMU assemblies from within EPD.Connect. You can use EPD.Connect to write directly to a CAMU assembly database file (`_db`).

Please note: EPD.Connect supports CADDs 5 Revision 6.1 and later.

Please note: The CADDs Interface can be purchased separately.

To enable EPD.Connect to access a CADDs CAMU database, you need to configure your workstation for normal CADDs CAMU installation. Refer to the *Concurrent Assembly Mock-Up User Guide and Menu Reference* for more information.

If you have Vault installed, see “CAD Options” on page 2-5.

Relocating the Initialization File

For CADDs, you normally set user-specific variables in your `.caddsrc-local` file and use a global `.caddsrc` file.

Please note: On UNIX platforms, EPD.Connect sources `.caddsrc` and `.caddsrc-local` from the home directory. If `.caddsrc` does not exist in the home directory, EPD.Connect sources it from the `/usr/apl/cadd/scripts/templates` directory. You must ensure that the `LANG`, `LC_CTYPE` and other local EPD.Connect variables are not reset in the `.caddsrc` file.

Please note: See *Installing CADDs 5i* and *Managing CADDs 5i* to learn about the `.caddsrc-local` and `.caddsrc` files.

Viewing Model Geometry

The Product View window (3D Viewer or EPD.Visualizer) can display CADDs tessellation files directly. See “Generating and Locating Graphic Binary Files” on page 4-2 to learn about viewing CADDs models in the 3D Viewer.

Setting CADD5 CVPATH

The CVPATH variable informs the application where various files are located and which directories must be used for searching files.

You must have the CVPATH variable set before invoking EPD.Connect. If you have not previously installed CADD5, you need to create the .caddsrc-local file in your user home directory. Otherwise, include the following in your existing .caddsrc-local file:

```
filename = .caddsrc-local
setenv CVPATH
# the =C indicates the create directory (that is, the new
# files should be created here.
$HOME/parts=C:'.:/usr/apl/cadd5:/usr/apl/cadd5/data:/usr
/                               ap1/cadd5/data/colortab
:/usr/ap1/cadd5/data/iges/lib:/usr/ap1/cadd5/data/vntab:
/
usr/ap1/cadd5/src/data:
/usr/ap1/cadd5/data/cvaec/hd/data'
```

Refer to page 8-15 for details on the CA_ASSYPREFIX and CA_READPREFIX variables required to activate assemblies through EPD.Connect menus.

Setting CAMU DB DAEMON Host Name

You must set the DB_DAEMON_HOST variable in order to specify the machine on which to run the ODB_SERVER process. An assembly-specific ODB_SERVER process is created on this machine each time you activate an assembly. Set this in your .caddsrc-local file in your \$HOME directory. A generic example is provided here.

```
setenv DB_DAEMON_HOST 'hostname'
```

Required Scripts for Vault and CADD5

If you are installing EPD.Connect on a server that runs Vault and also want to use CADD5, you must run one of the following UNIX scripts:

- `navinstall/navrefresh`—The EPD.Connect Database Installation utility. Run `navinstall` if you are installing Vault for the first time. Use `navrefresh` if you have already installed Vault and want to migrate from one revision to another.

These scripts:

- Create Oracle tables required for EPD.Connect.
- Create the EPD.Connect/Vault attribute views.
- Check for CAMU ADRAW rulebase support.

The file is specifically designed for the Vault server and enables you to store CADD5 related data on (and retrieve data from) the Vault.

Please note: You must already have downloaded the CADD5 revision from the product distribution CD-ROM. Otherwise, you must first download it with SLIC.

Installing EPD.Connect with CADD5 CAMU on the Same System as Vault

EPD.Connect requires certain information about the Vault database environment. If you are using CADD5, it also requires version information for that product. You specify the CADD5 version during loading.

Perform the following procedures for installing EPD.Connect exactly as described. If you do not, the database can be corrupted.

1. Load the software from the distribution media, specifying the CADD5 version to place on your local disk.
2. Run the `navinstall` script as specified in the following sections.

Order of Installation

If you plan to install EPD.Connect on the same system as Vault, perform your installations in the following order.

Vault	Distributed Vault	
Standalone Vault	With Distributed Object Directory (DOD)	With Distributed Vault (DV)
1. edminstall	1. edminstall	1. edminstall
2. ddinstall	2. edmdvinstall	2. edmdvinstall
3. navinstall	3. ddinstall	3. ddinstall
	4. dddvinstall	4. navinstall
	5. navinstall	

Changing the DOD

Before you run the `ddinstall` script, decide on the type of Vault or the environment for each Vault. If you later change your mind, you will need to reinstall everything, as in each of the following examples:

- You run `edmdvinstall` and choose that Vault to be the DOD, but you later decide that the vault should only be a Distributed Vault.
- You run `edmdvinstall` and choose that vault to be the Distributed Vault, but you later decide that the Vault should be DOD.

Changing the Location of the Distributed Vault

If you run `edminstall`, then `ddinstall`, and later decide to install Distributed Vault, you must rerun `ddinstall`. Rerunning `ddinstall` loses any binder information you may have.

Running navinstall

The `navinstall` file updates Oracle tables and installs Vault attribute rules, Vault views, and Vault attributes.

You must run the `navinstall` script twice. The first time, the Vault attribute rules are installed. The second time, the Vault attributes are installed.

The `navinstall` program is interactive. It asks you to accept or supply system data and prompts you to confirm that information.

To run `navinstall`, follow these steps. Default values are shown in brackets ([]). Press `Return` to accept them.

1. As root, invoke the following command:

```
# $EPD_HOME/install/navinstall
```

If the environment variable `LANG` is not set, it defaults to `C` and asks whether you want to accept `LANG=C`. If you reply `Yes`, it will continue processing. If you reply `No`, `navinstall` aborts. You can set the `LANG` variable and rerun.

```
NAV012E
*****
NAV012E Environment variable LANG is undefined
NAV012E
*****
NAV032P LANG will default to C (English), Continue
? [yes] yes
```

The database installation utility begins.

```
*****
* Database Installation Utility for EPD.Connect
*****
```

2. Press `Return` to accept the default values given for the following:

- a. Enter the user ID of your Optegra Vault Administration account
:edmapl

- b. Enter the directory where Optegra Vault is installed
[/opt/epd/dm/v60] :

- c. Enter the directory where EPD.Connect is installed
[/opt/epd/dm/v60] :

3. Verify the values and press `Return` to accept them, or enter `yes` and correct them.

```
*****
Installation details you have specified are :-

Optegra Vault Administration account : edmapl
Optegra Vault installation directory :
/opt/epd/dm/v60
```

```
EPD.Connect installation directory      :  
/opt/epd/dm/v60  
*****  
Do you want to re-enter any values [no] ? :  
4. Press Return to install the EPD.Connect Oracle tables; then press  
Return to confirm the default values, or supply new ones.  
  
OK to install the EPD.Connect Oracle Tables [yes] :  
  
Enter the Optegra Vault Oracle database manager  
userid [pdmadm] :  
Enter the Optegra Vault Oracle database manager  
password [pdmadm] :  
  
Enter the Optegra Vault Oracle IQF userid [pdmqf] :  
Enter the Optegra Vault Oracle IQF password [pdmqf]  
:  
  
Enter the Optegra Vault Batch Transfer Concurrency  
[6] :  
  
Creating New Tables in ORACLE Database.  
*****  
  
Processing completed.  
*****  
5. Press Return to install the Optegra Vault; then press Return to confirm  
the default values, or supply new ones.  
  
OK to install the EPD.Connect Optegra Vault Views  
[yes] :  
  
Enter the Optegra Vault Oracle database manager  
userid [pdmadm] :  
Enter the Optegra Vault Oracle database manager  
password [pdmadm] :  
  
Enter the Optegra Vault Oracle IQF userid [pdmqf] :  
Enter the Optegra Vault Oracle IQF password [pdmqf]  
:  
  
*****
```

Creating New Views and Synonyms in ORACLE
Database.

```
*****
Press Return to install DV support for Oracle IQF
user.
Install the DV support for Oracle IQF user [yes] :
*****
Granting SELECT to Optegra Vault Oracle IQF User.
*****
Processing completed.
*****
```

6. Press Return to decline installation of EPD.Connect DV Views.

```
OK to install the EPD.Connect DV Views [no] :
*****
Requested NOT to install EPD.Connect DV Views.
*****
Sun Microsystems Inc.SunOS 5.5.6 Generic August
1997
NAV025P OK to install the Navigator Optegra Vault
Attribute rules [yes]
:
NAV044I
*****
NAV044I The filename attribute rule already exists.
NAV044I
*****
NAV039I
*****
NAV039I Running edmrparser for attrrule.cg
NAV039I
*****
NAV040I
*****
NAV040I Running edmrparser for attrtypes.cg
NAV040I
*****
NAV041I
*****
NAV041I Running ldamlogic.
NAV041I
*****
./ldamlogic: EDM control table load for EDMVault

NAV043I
```

```
*****
NAV043I Successfully installed new rule.
NAV043I
*****
NAV005I
*****
NAV005I THE VAULT MUST BE RESTARTED FOR THIS RULE
TO BECOME ACTIVE
NAV005I THIS WILL BE REQUIRED WHEN INSTALLING THE
VAULT ATTRIBUTES.
NAV005I PLEASE RESTART THE VAULT BEFORE CONTINUING
NAV005I RE-RUN THE INSTALL SCRIPT and INSTALL
ATTRIBUTES ONLY.
NAV005I
*****
After this, the following message appears:
NAV034P OK to install EPD Interface to CADD55
[yes]:
```

Exit `navinstall` by entering No for this option.

After this you have to exit from `navinstall`.
Restart vault and again run `navinstall` till this
step by selecting "no" for all options. Then
following message will be displayed --

```
NAV033P OK to install the Navigator Optegra Vault
Attributes [yes] :yes
```

```
--->>> Existing documentation for this should be
retained.
<<<---
```

```
NAV006P Enter the EDMADMIN user password
[edmadmin] :
```

```
NAV042I
*****
NAV042I Signing on to the Optegra Vaults as
edmadmin.
NAV042I
*****
CDMSON000I Optegra - Version 6.0
```

```
*****  
CDMSON016I Sign on to EDM server ADBHUT completed  
successfully. You have 0 EDM message(s).
```

```
NAV010I
```

```
*****
```

```
NAV010I Adding mandatory attribute set and  
attributes.
```

```
NAV010I
```

```
*****
```

```
CAMAAS100I CONFIG_SET has been added.
```

```
CAMAAT100I COMPONENT-NAME has been  
added.
```

```
CAMAAT100I ITEM-NAME has been  
added.
```

```
CAMAAT100I REVISION has been  
added.
```

```
CAMAAT100I TYPE has been  
added.
```

```
CAMAAT100I PARENT-NAME has been  
added.
```

```
CAMAAT100I COMPONENT-ORDER has been  
added.
```

```
CAMAMA100I COMPONENT-NAME has been  
added.
```

```
CAMAMA100I ITEM-NAME has been  
added.
```

```
CAMAMA100I REVISION has been  
added.
```

```
CAMAMA100I TYPE has been  
added.
```

```
CAMAMA100I PARENT-NAME has been  
added.
```

```
CAMAMA100I COMPONENT-ORDER has been  
added.
```

```
CAMAAS100I CONFIG_OPTIONAL has been added.
```

```
NAV007I
```

```
*****
```

```
NAV007I Adding Optional Attributes.
```

```
NAV007I
```

```
*****
```

```
CAMAAT100I ATTRIBUTE has been  
added.
```

```
CAMAAT100I ORIENTATION has been
```

```
added.
CAMAAT100I GLOBAL-ORIENT has been
added.
CAMAAT100I ADRAW has been
added.
CAMAAT100I PARENT-RELATION has been
added.
CAMAAT100I ATTR-GROUP has been
added.
CAMAAT100I ATTR-DEFINITION has been
added.

NAV027I
*****
NAV027I Populating Attribute Sets.
NAV027I
*****
CAMAMA100I ORIENTATION has been
added.
CAMAMA100I GLOBAL-ORIENT has been
added.
CAMAMA100I ADRAW has been
added.
CAMAMA100I PARENT-RELATION has been
added.
CAMAMA100I ATTR-GROUP has been
added.
CAMAMA100I ATTR-DEFINITION has been
added.
CAMAMA100I ATTRIBUTE has been
added.

NAV008I
*****
NAV008I Adding config rules.
NAV008I
*****
NAV012I
*****
NAV012I Attribute Installation completed.
NAV012I
*****
```

After this, the following message appears:

```
NAV034P OK to install EPD Interface to CADD55  
[yes]:
```

Enter Yes for this option.

If you want to install CADD5 rulebase, select yes for the following option -- following is the sample log of CADD5 rulebase installation --

```
*****  
*      Installation Utility  
*****  
OK to add CADD5 Application Environment to Optegra  
Vault [yes] :  
  
Enter the Vault Oracle database manager userid  
[pdmdm] :  
Enter the Vault Oracle database manager password  
[pdmdm] :  
  
Checking New Tables in ORACLE Database .....  
Enter the Vault Oracle database manager userid  
[pdmdm] :  
Enter the Vault Oracle database manager password  
[pdmdm] :  
  
Checking New Tables in ORACLE Database .....  
  
Enter the Vault Oracle IQF userid [pdmqf] :  
Enter the Vault Oracle IQF password [pdmqf] :  
  
Creating New Tables and Views in ORACLE Database  
.....  
  
Toolkit Table creation Complete.  
  
Enter the Vault Oracle database manager userid  
[pdmdm] :  
Enter the Vault Oracle database manager password  
[pdmdm] :  
  
Adding CADD5 Application to Optegra Vault.
```

Addition of CADDs Application to Optegra Vault

OK to install the CADDs Vault Attributes [yes] :

Enter the EDMADMIN user password [edminadmin] :

Signing on to the Vault as edminadmin..

CDMSON000I Optegra

CDMSON016I Sign on to EDM server SARASWATI

completed successfully. You have 0 EDM message(s).

Adding CADDs attributes.

CAMARL100I CONFIG_RULE_DB1 has been added.

CAMARL100I CONFIG_RULE_PS1 has been added.

CAMARL100I CONFIG_OPT_RULE_DB1 has been added.

CAMARL100I CONFIG_OPT_RULE_PS1 has been added.

CDMSOF017I Sign off from EDM completed

successfully.

CADDs attributes added.

Optegra Interface for CADDs Installation
Complete.

NAV015I

NAV015I EPD.Connect Installation completed.

NAV015I

Processing completed.

EPD.Connect Installation completed.

Please note: You must have the appropriate CADDs support files in
the \$EPD_HOME/lib directory for navinstall to execute successfully.

Installing EPD.Connect with CADDs on a System with Remote Access to Vault

Please note: You must download the desired CADDs revision from the product distribution media (using SLIC) before running this script.

Installing EPD.Connect for the First Time

For a fresh installation, do the following:

To set up the Vault Client on UNIX, run the `edminstall` script. This script is run only for a fresh installation.

Please note: The `nsm.config` file located on the Vault server must contain your client machine name.

For information about accessing Vault from EPD.Connect, refer to the *Installing Vault and Locator*.

Please note: Installing Locator is not a prerequisite for installing and running EPD.Connect. Create the `pm.config` file and the `EDM.DEFAULTS` file to run EPD.Connect without Locator.

Upgrading an Installation

You must have the appropriate CADDs support files in the `$EPD_HOME/lib` directory for `navinstall` to execute successfully.

Accessing Vault from EPD.Connect

To access Vault from EPD.Connect, perform the following steps before invoking EPD.Connect:

1. Create a `pm.config` file in your `$HOME` directory. The `pm.config` file must contain a `RESOURCE` statement that maps to the process manager of the Vault you want to access.
2. Set the following environment either at the command line or in the `.login` or `.cshrc` file:

```
setenv ANSPATH $HOME/pm.config
```

Listing Parts from within Vault

To list files, parts, catalogs, and directories in a local directory other than \$HOME, set the following in your `.Edmgui` file:

```
Edmgui.DefaultEdmLocalPartDirectory: directory
Edmgui.DefaultEdmLocalFileDirectory: directory
Edmgui.DefaultEdmLocalDirDirectory: directory
Edmgui.DefaultEdmLocalCatDirectory: directory
```

Sizing the CADDs Windows

You can reduce the size of CADDs windows to see both the EPD.Connect windows and CADDs windows simultaneously. If you do not resize the CADDs display, it will cover the entire screen.

Use variables described in the template `.caddsrc` file (`/usr/apl/cadd/scripts/templates`) to size various CADDs windows.

You can also include the following script in your `.caddsrc-local` file (resident in your \$HOME directory). It uses the `xwininfo` command. The `xwininfo` command is located in different places on different machines. For example, on an HP machine it resides at `/usr/contrib/bin/x11`. On a Solaris machine it resides at `/usr/openwin/bin`.

Sample Script

```
# This file can be appended to your .caddsrc-local.
# Useful for resizing CADDs/CAMU window.
# Make sure there are no conflicts with your environment.

# Figure out the correct window size based on the
# resolution of the
# screen.
#
@ ScreenXSize=`xwininfo -root | grep Width | sed -e
`s/.*/`
@ ScreenYSize=`xwininfo -root | grep Height | sed -e
`s/.*/`
#
@ Factor=($ScreenXSize * 1000) / 1152
@ YFactor=($ScreenYSize * 1000) / 900
#
```

```

@ GraphXSize=( $ScreenXSize - (( $Factor * 133 ) / 1000)) -
20
@ GraphYSize=( $ScreenYSize - (( $Factor * 55 ) / 1000)) -
60
@ PictXSize=$GraphXSize
@ PictYSize=$GraphYSize
@ TextXSize=113
@ TextYSize=10
#
@ GraphXLoc=( $Factor * 133 ) / 1000
@ GraphYLoc=( $Factor * 55 ) / 1000
@ PictYLoc=$GraphYLoc + 20
@ TextYLoc=$ScreenYSize - 160
#
setenv CADDs_GRAP_GEOM
"${GraphXSize}x${GraphYSize}+${GraphXLoc}+${GraphYLoc}"
setenv CADDs_PICT_GEOM
"${PictXSize}x${PictYSize}+${GraphXLoc}+${PictYLoc}"
setenv CADDs_TEXT_GEOM
"${TextXSize}x${TextYSize}+${GraphXLoc}+${TextYLoc}"
#
set scalewindow = 1
if ( $scalewindow == 1 ) then
#
@ ScreenXSize=`xwininfo -root | grep Width | sed -e
`s/.*/`
@ ScreenYSize=`xwininfo -root | grep Height | sed -e
`s/.*/`
#
# cadd window size factor
#
if ( ! $?CVSIZE ) then
    setenv CVSIZE 0.7
#
    setenv CVSIZE 0.99
endif
#echo "    Menu Size Factor .... $CVSIZE"
@ Xsize = `echo "$ScreenXSize $CVSIZE" | awk
`{sum=$1*$2;printf "%.0d\n", sum}`
@ Ysize = `echo "$ScreenYSize $CVSIZE" | awk
`{sum=$1*$2;printf "%.0d\n", sum}`
#
setenv CV_UI_MENU_WIDTH $Xsize
setenv CV_UI_MENU_HEIGHT $Ysize
#
@ ScreenXSize = $Xsize
@ ScreenYSize = $Ysize
#
@ Factor=( $ScreenXSize * 1000 ) / 1152
@ YFactor=( $ScreenYSize * 1000 ) / 900
    
```

```
#
@ GraphXSize=($ScreenXSize - (($Factor * 133) / 1000)) -
20
@ GraphYSize=($ScreenYSize - (($Factor * 55) / 1000)) -
60
@ PictXSize=$GraphXSize
@ PictYSize=$GraphYSize
@ TextXSize=113
@ TextYSize=10
#
@ GraphXLoc=($Factor * 133) / 1000
@ GraphYLoc=($Factor * 55) / 1000
@ PictYLoc=$GraphYLoc + 20
@ TextYLoc=$ScreenYSize - 160
@ TextXLoc=$GraphXLoc
#
set CaddsRev="5"
if ( "$CaddsRev" == "5" ) then
    # @ TextXSize=135
    # 8 pixels per character
    @ TextXSize= ($GraphXSize * 100) / 790
    @ TextYSize=5
    @ TextYLoc=$ScreenYSize - 70
    @ TextXLoc=$GraphXLoc - 3
endif
#
setenv CADDs_GRAP_GEOM
"${GraphXSize}x${GraphYSize}+${GraphXLoc}+${GraphYLoc}"

setenv CADDs_PICT_GEOM
"${PictXSize}x${PictYSize}+${GraphXLoc}+${PictYLoc}"
setenv CADDs_TEXT_GEOM
"${TextXSize}x${TextYSize}+${GraphXLoc}+${TextYLoc}"
#
endif
```

Please note: You can append the
\$EPD_HOME/.caddrsc-local-win file contents to your
\$HOME/.caddsrc-local file in order to display CADDs in a more
suitably sized window area.

Considerations

- Interoperability from the Assembly Tree (Product Structure window) and CAMU (CADDs Graphics window) is valid only when single-component selection mode is active.
- CADDs/CAMU must be started from the EPD.Connect Tools menu.

Configuring the EPD.Connect Interface for CADDs

For information on configuring the EPD.Connect interface for CADDs, see “Configuring for CADDs” on page 5-2.

Configuring EPD.Connect Interfaces

This chapter describes how to configure EPD interfaces. The following topics are presented:

- Configuring for CADD5
- Configuring for MEDUSA
- Configuring for Pro/ENGINEER
- Configuring for CATIA
- Configuring for STEP AP203
- Shared Application Variable Settings
- Repository Files for EPD.Connect Interfaces

Please note: See Appendix C, “EPD.Connect Preconfiguration Checklist” for items to verify before attempting configuration tasks.

Configuring for CADD5

Configuring the Vault Server

The EPD.Connect interface to CADD5 is configured when `navinstall` is run for EPD.Connect on the Vault.

Please note: By default, the `navinstall` script automatically installs the Optegra Interface for CADD5.

The instructions for separately configuring the Vault server on UNIX and Windows NT follow.

UNIX

Please note: Do not “tee” the output from `navinstall`, `navrefresh`, or `cadinst`.

If you have already run `navinstall` or `navrefresh` but did not complete the installation, run the following as Optegra Vault administrator:

```
$EPD_HOME/install/cadinst
*****
* Installation Utility
*****
OK to add CADD5 Application Environment to Optegra
Vault [yes] :

Enter the Vault Oracle database manager userid
[pdmdm] :
Enter the Vault Oracle database manager password
[pdmdm] :

Checking New Tables in ORACLE Database .....
Enter the Vault Oracle database manager userid
[pdmdm] :
Enter the Vault Oracle database manager password
[pdmdm] :

Checking New Tables in ORACLE Database .....

Enter the Vault Oracle IQF userid [pdmqf] :
Enter the Vault Oracle IQF password [pdmqf] :
```

Creating New Tables and Views in ORACLE Database
.....

Toolkit Table creation Complete.

Enter the Vault Oracle database manager userid

[pdmdm] :

Enter the Vault Oracle database manager password

[pdmdm] :

Adding CADD5 Application to Optegra Vault.

Addition of CADD5 Application to optegra Vault

OK to install the CADD5 Vault Attributes [yes] :

Enter the EDMADMIN user password [edmadmin] :

Signing on to the Vault as edmadmin..

CDMSON000I Optegra

CDMSON016I Sign on to EDM server SARASWATI
completedsuccessfully.Youhave0EDMmessage(s).

Adding CADD5 attributes.

CAMARL100I CONFIG_RULE_DB1 has been added.

CAMARL100I CONFIG_RULE_PS1 has been added.

CAMARL100I CONFIG_OPT_RULE_DB1 has been added.

CAMARL100I CONFIG_OPT_RULE_PS1 has been added.

CDMSOF017I Sign off from EDM completed
successfully.

CADD5 attributes added.

Optegra Interface for CADD5 Installation Complete.

EPD.Connect Installation Completed.

Windows NT

Follow these steps:

1. Set the following variables in your system's environment:

```
set LANG=enu
set PERL_PATH=%EPD_HOME%\PERL5\BIN\MSWIN32-X86\
PERL-CONSOLE
```

2. Log in as the Optegra Vault Administrator.
3. Change to the %EPD_HOME% installation directory.
4. Run the install script using the following command:

```
%PERL_PATH% cadinst
```

Configuring the Vault Client

Generic behavior for the CADD5 interface is controlled by the variables defined in the `epdconn.ini` file, as mentioned in the preceding section. Application-specific behavior is controlled by the settings in the repository file `cadd5drb.ini`.

Please note: See also “Shared Application Variable Settings” on page 5-30.

cadd5drb.ini — Repository for CADD5 Interface

The default location for this file is `$EPD_HOME/data/reposit/cadd5drb.ini`. The file contains the following variables:

```
ENV_MSG=$EPD_HOME/data/reposit/$AWLANG/cadd5msg.txt
—The location for the message repository file.

CHECK_TD=NO/YES — If YES, checks the concurrency of the _pd and
_td files for the CADD5 part.

PD_MANDATORY=YES — If YES, a _pd file is mandatory for storing the
CADD5 part.

TD_MANDATORY=NO — If YES, a _td file is mandatory for storing the
CADD5 part.

FD_MANDATORY=NO — If YES, an _fd file is mandatory for storing the
CADD5 part.
```

`STORE_TVFS=YES` — Determines whether or not to store these CADDs-related files.

`STORE_TEXT=YES` — Determines whether or not to store these CADDs-related files.

`STORE_REDLINES=YES` — Determines whether or not to store these CADDs-related files.

`STORE_ADRAWS=YES` — Determines whether or not to store the ADRAWS.

`NAME_CONVENTION=convert` — Related to the conversion of local file names. Note: Do not change this setting.

The following variables are for defining the `PREPROCESS` and `POSTPROCESS` executables while executing the rulebase commands:

```
STORE_PREPROCESS=  
UPDATE_PREPROCESS=  
GET_PREPROCESS=  
READ_PREPROCESS=  
LIST_PREPROCESS=  
STORE_POSTPROCESS=  
UPDATE_POSTPROCESS=  
GET_POSTPROCESS=  
READ_POSTPROCESS=  
LIST_POSTPROCESS=
```

For more information, see “Preprocess and Postprocess Variables for the EPD.Connect Interfaces” on page 5-36.

Using the Vault Command Line Interface

Set the following variables in your environment to use the EPD.Connect interfaces with the Vault command line interface:

- Set `EPD_HOME` to the directory where Optegra is installed
- If you do not have a customized `cvepd.ini` file, set `CVEPD_INIT` to " "
- Set `CVEPD_MAIN_CONFIG` to `$EPD_HOME/data/reposit/epdconn.ini`
- Set `EPD_TEMP` to `/usr/tmp/nav`
- Set `CA_TEMP` to `/usr/tmp/nav` (The directory for temporary usage)

- Set `ANSPATH` to the location of the `pm.config` file
- Set `DB_DAEMON_HOST` to the appropriate server machine in the event you want to work with CAMU assemblies

Configuring for MEDUSA

Please note: The EPD Interface to MEDUSA is a licensed product and can be purchased separately.

The executable file for EPD Interface to MEDUSA is `medusaif`. It is loaded in the `$(EPD_HOME)/bin` directory and specified when EPD.Connect is loaded. The executable is called whenever a Vault client command is executed for the MEDUSA-generated objects.

The `$(EPD_HOME)/data/reposit/medusarb.ini` file contains a list of variables you set to enable the EPD Interface to MEDUSA through EPD.Connect. You must edit `medusarb.ini` so that these variables are set to proper values for your configuration. For details see “Configuring the Vault UNIX Client” on page 5-8.

Configuring the Vault Server

UNIX

After you have loaded Optegra Vault and the EPD Interface to MEDUSA, type the following on the Vault server:

1. `su <Vault Admin>`
2. `cd $(EPD_HOME)/install`
3. `./medinst`

Please note: You must be logged in as the Optegra Vault Administrator to run the `medinst` script.

The `medinst` script adds the MEDUSA application environment to EPD.Connect and adds MEDUSA attributes and rules to the Vault.

To set up MEDUSA support on the Vault, run the following as Vault administrator only after you have run either `navinstall` or `navrefresh`:

```
$(EPD_HOME)/install/medinst
```

```
*****
*  Installation Utility
*****
Enter the user id of your Vault Administration
account : edmapl

OK to add MEDUSA Application Environment to Optegra
Vault [yes] :

Enter the Vault Oracle database manager userid
[pdmdm] :
Enter the Vault Oracle database manager password
[pdmdm] :

Checking New Tables in ORACLE Database .....
New tables already exist.

Enter the Vault Oracle database manager userid
[pdmdm] :
Enter the Vault Oracle database manager password
[pdmdm] :

Adding MEDUSA Application to Optegra Vault.

OK to install the MEDUSA Vault Attributes [yes] :

Enter the EDMADMIN user password [edmadmin] :

Signing on to the Vault as edmadmin
CDMSON000I Optegra

CDMSON016I Sign on to EDM server AMEY completed
successfully. You have 2 EDM message(s).
```

```
Adding MEDUSA attributes.  
CAMARL100I MEDUSA_SET has been added.  
CAMARL100I MEDUSA-DEPENDENT has been added.  
CAMARL100I MEDUSA_RULE_1 has been added.  
CAMARL100I MEDUSA_RULE_2 has been added.  
CAMARL100I MEDUSA_RULE_3 has been added.  
CAMARL100I MEDUSA_RULE_4 has been added.  
CDMSOF017I Sign off from EDM completed  
successfully.  
MEDUSA attributes added.
```

Optegra Interface for MEDUSA Installation Complete.

Windows NT

Follow these steps:

1. Set the following variables in your system's environment:

```
LANG=enu  
PERL_PATH=%EPD_HOME%\PERL5\BIN\MSWIN32-X86\PERL-CONS  
OLE
```

2. Log in as the Optegra Vault Administrator.
3. Change to the %EPD_HOME% installation directory.
4. Run the install script using the following command:

```
%EPD_HOME%\PERL5\BIN\MSWIN32-X86\PERL-CONSOLE  
medinst
```

Configuring the Vault UNIX Client

Generic behavior for the MEDUSA interface is controlled by the variables defined in the `epdconn.ini` file. These are listed in “Shared Application Variable Settings” on page 5-30. Application-specific behavior is controlled by the settings in the repository file, `medusarb.ini`. The default location for this file is `$(EPD_HOME)/data/reposit/medusarb.ini`.

medusarb.ini — Repository for the MEDUSA Interface

This file contains the following variables:

```
ENV_MSG=$(EPD_HOME)/data/reposit/$AWLANG/medmsg.txt—  
The location for the message repository file.
```

`MEDPROJ_PATH=$HOME/parts` — The Directory where the MEDUSA objects are available.

`DDL_PATH='[Path for ddl.bin including filename]'` — The path for the MEDUSA application file, `ddl.bin`, including the file name.

`STORE_PREPROCESS=`

`UPDATE_PREPROCESS=`

`GET_PREPROCESS=`

`READ_PREPROCESS=`

`REPLACE_PREPROCESS=`

`LIST_PREPROCESS=`

`STORE_POSTPROCESS=`

`UPDATE_POSTPROCESS=`

`GET_POSTPROCESS=`

`READ_POSTPROCESS=`

`REPLACE_POSTPROCESS=`

`LIST_POSTPROCESS=`

These variables are set to `NULL` during the installation. They can be set to the customized `PREPROCESS` and `POSTPROCESS` executables, so that they are called whenever the respective Vault command is executed.

The next tables list and describe environment variables that you must set in either your environment or the appropriate `.ini` file before you launch EPD.Connect or MEDUSA.

The following variables are required by the MEDUSA interface rulebase and are specified in `medusarb.ini`.

Variable	Value	Used by
<code>DDL_PATH</code>	Path for <code>ddl.bin</code> , including the file name	MEDUSA Interface
<code>MEDPROJ_PATH</code>	User's MEDUSA project directory	MEDUSA Interface

The following variables are required when the MEDUSA application is customized to work with the EPD Interface to MEDUSA. Set them in your environment.

Variable	Value	Used by
EPD_HOME	EPD.Connect Directory	EPD.Connect, Vault
DISPLAY	Same as EPD.Connect display	EPD.Connect messaging
CMOM_DISPLAY	Same as EPD.Connect display	EPD.Connect messaging
CMOM_DOMAIN	Same as EPD.Connect Domain	EPD.Connect messaging
CMOM_CLIENT_T IMEOUT	300	EPD.Connect messaging

Customizing the MEDUSA Application Menus

To perform Vault transfer operations from within the MEDUSA environment, you must customize the MEDUSA application. You can also perform Vault transfer commands within EPD.Connect.

The EPD Interface to MEDUSA provides integration scripts for embedding Vault commands in the MEDUSA application. You can invoke these scripts from a customized menu created in the MEDUSA application. The arguments needed for these scripts should be supplied by the customized menus. Each script is called as follows:

```
scriptname.pl [arg1,] [arg2,] [arg3,]
```

The following table lists integration script file names and their arguments.

File Name	Arguments	Definition
Signon.pl	arg1: user-id	Optegra user ID
Signoff.pl	none	
Store.pl	arg1: Application	Application type, that is, MEDUSA
	arg2: Selection Name	Object's name in the Vault
	arg3: Local Name	Object's local file name
Get.pl	arg1: Application	Application type, that is, MEDUSA
	arg2: Selection Name	Object's name in the Vault
Read.pl	arg1: Application	Application type, that is, MEDUSA
	arg2: Selection Name	Object's name in the Vault
	arg3: Revision	Object's revision number
Update.pl	arg1: Application	Application type, that is, MEDUSA
	arg2: Selection Name	Object's name in the Vault
Replace.pl	arg1: Application	Application type, that is, MEDUSA
	arg2: Selection Name	Object's name in the Vault

These scripts can be embedded in the MEDUSA application by adding a new menu for the Vault operations. This task is part of customizing MEDUSA applications and is discussed in the *MEDUSA Customization Guide*.

These integration scripts communicate with Vault through EPD.Connect. For this reason, EPD.Connect must be running on your Client machine. The scripts use EPD.Connect messaging to communicate with EPD.Connect. To enable this, you must set the following variables in your MEDUSA environment:

- DISPLAY
- CMOM_DISPLAY

- CMOM_DOMAIN
- CMOM_CLIENT_TIMEOUT

These variables are described in “Configuring the Vault UNIX Client” on page 5-8. Also, you need to set the rest of the variables listed in the `medusarb.ini` file.

Using the Vault Command Line Interface

Set the following variables in your environment to use the EPD.Connect interfaces with the Vault command line interface:

- Set `EPD_HOME` to the directory where Optegra is installed.
- If you do not have a customized `cvepd.ini` file, set `CVEPD_INIT` to `" "`.
- Set `CVEPD_MAIN_CONFIG` to `$(EPD_HOME)/data/reposit/epdconn.ini`.
- Set `EPD_TEMP` to `/usr/tmp/nav`.
- Set `CA_TEMP` to `/usr/tmp/nav` (the directory for temporary usage).
- Set `ANSPATH` to the location of the `pm.config` file.

Local Use of EPD Interfaces

To use the EPD Interface for MEDUSA, disable the existing Vault support. For more information, refer to the section “Connecting to the Vault” on page 2-10.

If you have not disabled the Vault support, and you want EPD.Connect to list and recognize the application interface types, the first entry in the `pm.config` must be for a Vault that you have configured for the installed interfaces.

Configuring for Pro/ENGINEER

The executable file for the EPD Interface to Pro/ENGINEER is named `proeif` and is located in the `$(EPD_HOME)/bin` directory. This executable is called whenever a Vault command is executed for the Pro/ENGINEER-generated objects.

The EPD Interface to Pro/ENGINEER supports Releases 11 through 21 and 2000i of Pro/ENGINEER.

Please note: The EPD Interface to Pro/ENGINEER is a licensed product and can be purchased separately.

Configuring the Vault Server

Instructions for configuring the Vault server on UNIX and Windows NT follow.

UNIX

After you have loaded Optegra Vault, perform the following steps on the Vault to configure the Vault server:

1. Log in as Optegra Vault Administrator.
2. Set the following variable in your system's environment path:
`$EPD_HOME/PERL5/bin`
3. Set an appropriate value for the LANG variable depending on the locale.
4. Type `cd $EPD_HOME/install`.
5. Type `proeinst`.

Please note: Do not "tee" the output from `proeinst` script.

These commands add the Pro/ENGINEER Application Environment to EPD.Connect and add Pro/ENGINEER attributes and rules to the Vault.

To set up Pro/ENGINEER support on the Vault, run the following as Vault administrator only after you have run either `navinstall` or `navrefresh`:

```
$EPD_HOME/install/proeinst
```

```
*****
*   Installation Utility
*****
Enter the user id of your Vault Administration
account   :edmapl

OK to add Pro/E Application Environment to Optegra
Vault [yes] :
```

```
Enter the Vault Oracle database manager userid  
[pdmdm] :  
Enter the Vault Oracle database manager password  
[pdmdm] :
```

```
Checking New Tables in ORACLE Database .....
```

```
New tables already exist.  
Enter the Vault Oracle database manager userid  
[pdmdm] :  
Enter the Vault Oracle database manager password  
[pdmdm] :
```

```
Adding Pro/E Application to Optegra Vault.
```

```
Addition of Pro/E Application to optegra Vault  
Complete.
```

```
OK to install the Pro/E Vault Attributes [yes] :
```

```
Enter the EDMADMIN user password [edmadmin] :
```

```
Signing on to the Vault as edmadmin
```

```
CDMSON000I Optegra
```

```
CDMSON016I Sign on to EDM server AMEY completed  
successfully. You have 2 EDM message(s).
```

```
Adding Pro/E attributes.  
CAMARL100I CONFIG_RULE_ASM has been added.  
CAMARL100I CONFIG_OPT_RULE_ASM has been added.  
CDMSOF017I Sign off from EDM completed  
successfully.  
Pro/E attributes added.
```

```
Optegra Interface for Pro/E Installation Complete.
```

Windows NT

Follow these steps to configure the Vault server:

1. Log in as the Optegra Vault Administrator.
2. Set the following variables in your system's environment:

```
LANG=enu  
PERL_PATH=%EPD_HOME%\PERL5\bin
```
3. Change to the %EPD_HOME%\install installation directory.
4. Run the install script using the following command:

```
%EPD_HOME%\install> %PERL_PATH%/PERL-CONSOLE  
proeinst
```

Migration of Existing Interface Data

The Migration tool is available and required only in UNIX environments. Use it to enable the data stored in versions of the Optegra Interface to Pro/ENGINEER earlier than Revision 16.

Previous versions of the EPD Interface to Pro/ENGINEER stored all the Pro/ENGINEER objects as parts_type=PTCOBJ in the dm_part_directory table in the Vault.

The EPD Interface to Pro/ENGINEER stores Pro/ENGINEER objects as part_types as listed here.

Pro/ENGINEER Objects	Part Types
ASSEMBLY	Stored as PTCASM
PART	Stored as PTCPRT
DRAWING	Stored as PTCDRW
LAYOUT	Stored as PTCLAY
FORMAT	Stored as PTCFRM

Please note: The necessary migration is automatically performed when the Optegra Vault is installed.

Configuring the Client

This section describes how to configure the EPD Interface to Pro/ENGINEER after it is loaded on a Vault client. You can configure a user account individually or globally. This interface refers to the initialization file named `ptcarb.ini`. The path for this file can be set using the environment variable `PTCRB_REPOSIT`. The default setting for this is `$EPD_HOME/data/reposit/ptcarb.ini`. This file contains some tokens that are referred to by the interface at run time.

You can configure the interface rulebase by editing the `ptcarb.ini` file. You can control various factors from the `ptcarb.ini` file. You can also set the preprocess and postprocess variables by editing this file.

For local configuration and customization, edit the `ptcarb.ini` file in your home directory. Set the environment variable `PTCRB_REPOSIT` to point to the customized `PTCARB.INI`.

Please note: On UNIX, set the `PTCRB_REPOSIT` variable in the user environment. This variable is not needed when the interface is running in Windows. The interface looks for the `ptcarb.ini` file in the Windows directory.

Generic behavior for the Pro/ENGINEER interface is controlled by the variables defined in the `epdconn.ini` file.

Extracting the Orientation Information from the Pro/ENGINEER Assembly

You can extract the orientation information for the components of the Pro/ENGINEER assembly using the executable `asm2psn` that resides in `$EPD_HOME/bin` directory.

On the UNIX platform, `asm2psn` works only on Solaris 2.6 that has the “Creator3D” graphics card. You must also install the OS patch “105591-02”, the OpenGL library, and the `xgl` library. Also, `asm2psn` works only on Pro/ENGINEER assemblies that are created in Pro/ENGINEER Release 20 and onward.

In case you are working with Windows NT, you must install the OpenGL library. (It gets installed with Windows NT. The DLLs used are `opengl32.dll` and `glu32.dll`). Also, `asm2psn` works only on

Pro/ENGINEER assemblies that are created in Pro/ENGINEER Release 20 and onward.

Guidelines for Installation

The instructions for installing `asm2psn` on the UNIX and Windows NT platforms follow:

UNIX:

1. Take a backup of the existing `$EPD_HOME/bin/asm2ps` binary.
2. Copy `$EPD_HOME/bin/asm2psn` to `$EPD_HOME/bin/asm2ps`.
3. Before you start EPD.Connect, set the `LD_LIBRARY_PATH` so that it contains the following directory:

`/opt/SUNWits/Graphics-sw/xgl/lib`

This is the directory that contains the OpenGL and the xgl libraries.

Windows NT:

1. Take a backup of the existing `%EPD_HOME%/bin/asm2ps.exe` binary.
2. Copy `%EPD_HOME%/bin/asm2psn.exe` to `%EPD_HOME%/bin/asm2ps.exe`.

ptcarb.ini — Repository for the Pro/ENGINEER Interface

Application-specific behavior is controlled by the settings in the repository file, `ptcarb.ini`. The default location for this file is `$EPD_HOME/data/reposit/ptcarb.ini`.

The file contains the following variables:

`ENV_MSG=$EPD_HOME/data/reposit/$AWLANG/ptcrbmsg.txt`
— The location for the message repository file.

`ALLOW_UPDATE_PRT_WO_DRW=YES/NO` — If YES, allows update/replace, even if the associated member DRW file is not available.

`CA_PROE_CLEAN_ALL=1/0` — If 1, purge all the versions of the stored object after store/replace.

`PROE_PATH=$HOME/parts` — The path for the directory where Pro/ENGINEER objects are available.

The variables listed next are for defining the `PREPROCESS` and `POSTPROCESS` executables while executing the rulebase commands.

Please note: The Pro/ENGINEER interface needs `STORE_PREPROCESS` to be set to the executable `$EPD_HOME/bin/strpp`. To execute another executable, execute the binary `strpp` after the execution of the custom binary.

```
STORE_PREPROCESS=$EPD_HOME/bin/strpp
UPDATE_PREPROCESS=
GET_PREPROCESS=
READ_PREPROCESS=
LIST_PREPROCESS=
STORE_POSTPROCESS=
UPDATE_POSTPROCESS=
GET_POSTPROCESS=
READ_POSTPROCESS=
LIST_POSTPROCESS=
```

Please note: See also “Shared Application Variable Settings” on page 5-30.

Using the Vault Command Line Interface

Set the following variables in your environment to use the EPD.Connect interfaces with the Vault command line interface:

- Set `EPD_HOME` to the directory where Optegra is installed.
- If you do not have a customized `cvepd.ini` file, set `CVEPD_INIT` to `" "`.
- Set `CVEPD_MAIN_CONFIG` to `$EPD_HOME/data/reposit/epdconn.ini`.
- Set `EPD_TEMP` to `/usr/tmp/nav`.
- Set `CA_TEMP` to `/usr/tmp/nav` (the directory for temporary usage).
- Set `ANSPATH` to the location of the `pm.config` file.

Local Use of EPD Interfaces

To use the EPD Interface for Pro/ENGINEER, disable the existing Vault support. For more information, refer to the section “Connecting to the Vault” on page 2-8.

If you have not disabled the Vault support, and you want EPD.Connect to list and recognize the application interface types, then the first entry in the `pm.config` must be for a Vault that you have configured for the installed interfaces.

Configuring for CATIA

Please note: The EPD Interface to CATIA is a licensed product and can be purchased separately.

Configuring Vault Server

Instructions for configuring the Vault server on UNIX and Windows NT follow.

UNIX

After you have loaded the Optegra Vault software, follow these steps on the Vault:

1. Log in to the Optegra server as Optegra Vault Administrator.
2. Type `cd $EPD_HOME/install`
3. Type `./catinst`

Please note: Do not “tee” the output from the `catinst` script.

This script adds the CATIA Application Environment to EPD.Connect and CATIA attributes and rules to the Vault.

If you have already run `navinstall` or `navrefresh` but did not complete the installation, run the following as Vault administrator:

```
$EPD_HOME/install/catinst
*****
* Installation Utility
*****
```

Enter the user id of your Vault Administration
account : **edmap1**

OK to add CATIA Application Environment to Optegra
Vault [yes] :

Enter the Vault Oracle database manager userid
[pdmdm] :

Enter the Vault Oracle database manager password
[pdmdm] :

Checking New Tables in ORACLE Database
New tables already exist.

Enter the Vault Oracle database manager userid
[pdmdm] :

Enter the Vault Oracle database manager password
[pdmdm] :

Adding CATIA Application to Optegra Vault.
OK to install the CATIA Vault Attributes [yes] :

Enter the EDMADMIN user password [edmadmin] :

Signing on to the Vault as edmadmin
CDMSON000I Optegra

CDMSON016I Sign on to EDM server SHWETHA completed
successfully. You have 6 EDM message(s).

Adding CATIA attributes.
CAMARL100I CONFIG_RULE_CATASSY has been added.
CAMARL100I CONFIG_OPT_RULE_CATASSY has been added.
CDMSOF017I Sign off from EDM completed
successfully.

CATIA attributes added.

Optegra Interface for CATIA Installation Complete.

Windows NT

Follow these steps:

1. Set the following variables in your system's environment:

```
LANG=enu
PERL_PATH=%EPD_HOME%\PERL5\BIN\MSWIN32-X86\
PERL-CONSOLE
```

2. Log in as the Optegra Vault Administrator.
3. Change to the %EPD_HOME% installation directory.
4. Run the install script using the following command:

```
%EPD_HOME%\PERL5\BIN\MSWIN32-X86\PERL-CONSOLE
catinst
```

Configuring the Vault Client

Generic behavior for the CATIA interface is controlled by the variables defined in the `epdconn.ini` file. These are listed in “Shared Application Variable Settings” on page 5-30.

catiarb.ini — Repository for the CATIA Interface

Application-specific behavior is controlled by the settings in the repository file, `catiarb.ini`. The default location for this file is `$EPD_HOME/data/reposit/catiarb.ini`.

The file contains the following variables:

```
ENV_MSG=$EPD_HOME/data/reposit/$AWLANG/catiamsg.txt
— The location for the message repository file.
CVDPA_REV_CATASSY=0 — This is used by the extract script.
```

The variables listed next are for defining the `PREPROCESS` and `POSTPROCESS` executables while executing the rulebase commands:

```
STORE_PREPROCESS=
UPDATE_PREPROCESS=
GET_PREPROCESS=
READ_PREPROCESS=
REPLACE_PREPROCESS=
```

```
LIST_PREPROCESS=  
STORE_POSTPROCESS=  
UPDATE_POSTPROCESS=  
GET_POSTPROCESS=  
READ_POSTPROCESS=  
LIST_POSTPROCESS=  
REPLACE_POSTPROCESS=
```

Please note: See also “Shared Application Variable Settings” on page 5-30.

Using the Vault Command Line Interface

Set the following variables in your environment to use the EPD.Connect interfaces with the Vault command line interface:

- Set `EPD_HOME` to the directory where Optegra is installed.
- If you do not have a customized `cvepd.ini` file, set `CVEPD_INIT` to `" "`.
- Set `CVEPD_MAIN_CONFIG` to `$(EPD_HOME)/data/reposit/epdconn.ini`.
- Set `EPD_TEMP` to `/usr/tmp/nav`.
- Set `CA_TEMP` to `/usr/tmp/nav` (the directory for temporary usage).
- Set `ANSPATH` to the location of the `pm.config` file.

Local Use of EPD Interfaces

To use the EPD Interface for CATIA, disable the existing Vault support. For more information, refer to the section “Connecting to the Vault” on page 2-8.

If you have not disabled the Vault support, and you want EPD.Connect to list and recognize the application interface types, the first entry in the `pm.config` must be for a Vault that you have configured for the installed interfaces.

Configuring for STEP AP203

The EPD Interface for STEP AP203 (hereafter STEP Interface) is installed as an option to EPD.Connect.

Configuring the Vault Server

Instructions for configuring the Vault server on UNIX and Windows NT follow.

UNIX

After you have loaded Optegra Vault and the STEP Interface, type the following on the Vault server:

1. `su <Optegra Vault Administrator>`
2. `cd $EPD_HOME/install`
3. `./stepinst`

The `stepinst` installation script adds the STEP application environment to EPD.Connect.

Please note: Do not “tee” the output from the `stepinst` script.

In addition to the copyright notice, the following messages are displayed:

```
TRI002P Enter the user id of your Vault
Administration account:edmapl
TRI702P OK to add STEP Application Environment
to Optegra Vault [yes]:
TRI008P Enter the Vault Oracle database manager
userid [pdmdm]:
TRI009P Enter the Vault Oracle database manager
password [pdmdm]:
TRI152I Checking New Tables in ORACLE Database.....
TRI169I New tables already exist.
TRI008P Enter the Vault Oracle database manager
userid [pdmdm]:
TRI009P Enter the Vault Oracle database manager
password [pdmdm]:
TRI753I Adding STEP Application to Optegra Vault.
TRI705P OK to install the STEP Vault Attributes
[yes]:
TRI012P Enter the EDMADMIN user password
[edmadmin]:
TRI013I Signing on to the Vault as edmadmin....
CDMSON016I Sign on to Vault server AMEY completed
successfully. You have
200 Vault message(s).
```

```
TRI791I Adding STEP attributes.
CAMARL100I CONFIG_RULE_STEP has been added.
CAMARL100I CONFIG_OPT_RULE_STEP has been added.
CDMSOF017I Sign off from Vault completed
successfully.
TRI792I STEP attributes added.
TRI711I Optegra Interface for STEP Installation
Complete.
*****
* Installation Utility
*****
Enter the user id of your Vault Administration
account :edmap1

OK to add STEP Application Environment to Optegra
Vault [yes] :

Enter the Vault Oracle database manager userid
[pdmdm] :
Enter the Vault Oracle database manager password
[pdmdm] :

Checking New Tables in ORACLE Database .....

New tables already exist.

Enter the Vault Oracle database manager userid
[pdmdm] :
Enter the Vault Oracle database manager password
[pdmdm] :

Adding STEP Application to Optegra Vault.

OK to install the STEP Vault Attributes [yes] :

Enter the EDMADMIN user password [edmadmin] :

Signing on to the Vault as edmadmin..

CDMSON016I Sign on to EDM server AMEY completed
successfully. You have 2 EDM message(s).

Adding STEP attributes.
CAMARL100I CONFIG_RULE_STEP has been added.
```

```
CAMARL100I CONFIG_OPT_RULE_STEP has been added.  
CDMSOF017I Sign off from EDM completed  
successfully.
```

```
STEP attributes added.
```

```
Optegra Interface for STEP Installation Complete.
```

Windows NT

Follow these steps:

1. Set the following variables in your system's environment:

```
LANG=enu  
PERL_PATH=%EPD_HOME%\PERL5\BIN\MSWIN32-X86\  
PERL-CONSOLE
```

2. Log in as the Optegra Vault Administrator.
3. Change to the %EPD_HOME% installation directory.
4. Run the install script using the following command:

```
%EPD_HOME%\PERL5\BIN\MSWIN32-X86  
\PERL-CONSOLE stepinst
```

Configuring the Vault UNIX Client

Generic behavior for the STEP interface is controlled by the variables defined in the `epdconn.ini` file. These are listed in “Shared Application Variable Settings” on page 5-30.

Please note: The default for the `CVCREATESTEP` variable is `${HOME}/step`. To use this default setting, you need to create a `step` directory in your `${HOME}`, if it does not already exist. You can change this setting to use any other directory as a repository for your STEP objects.

Set the following variable in your environment:

```
Set STEP_PVS_LICENSE_FILE to  
/usr/CVswlm/theorem/theorem.licenses
```

stepb.ini — Repository for the STEP Interface

Application-specific behavior is controlled by the settings in the repository file, `steprb.ini`. The default location for this file is `$EPD_HOME/data/reposit/steprb.ini`.

The file contains the following variables:

- `ENV_MSG=$EPD_HOME/data/reposit/$AWLANG/steprbmsg.txt` — The location for the message repository file.
- `DB_HASH_REPO_INIT=$EPD_HOME/data/step/utils/InitHashDB.pl` — The location of the DB initialization file.
- `EDM_STEP_BLANK=_` — The blank character (` `) in the local filename is replaced by the underscore character (`_`) for the selection name.
- `CVDPA_REV_STEP=0` — This is used in the extract script.
- `CVCREATESTEP=$HOME/parts` — The step object directory.
- `HASHREPO=$EPD_TEMP/hashrepo` — The step hash repository. (Do not change this variable).
- `EXPORT_DEF=$STEP_DIR/custexp/export_def` — The file defining applications allowing geometry conversion via export-interface.
- `UNIT=mm` — Enter unit (mm, inch).
- `USE_SHORTNAMES=Y` — se short names for target STEP files.
- `CADDS_AP203_TRANSLATOR_DIR=/usr/apl/cvuts/scripts` — Set `CADDS_AP203_TRANSLATOR_DIR` to the location of `getap203/putap203` translator scripts:
 - `GETAP203_INI=$CA_DATA/reposit/getap203.ini`
 - `PUTAP203_INI=$CA_DATA/reposit/putap203.ini`
- `TSC_PROG_FILE=$HOME/progressfile` — The location of the output sent to the log file for the STEP viewer.
- `CAMU_ITEM_LIST=$HOME/camulist.lst` — The location of the output sent to the CAMU item list file.
- `STEPPVS_LICENSE_FILE=/usr/CVswlm/theorem/theorem.license s` — The location of the theorem license file for the use of `stepis203_pvs`.

The variables listed next are for defining the `PREPROCESS` and `POSTPROCESS` executables while executing the rulebase commands:

```
STORE_PREPROCESS=  
UPDATE_PREPROCESS=
```

```
GET_PREPROCESS=  
READ_PREPROCESS=  
REPLACE_PREPROCESS=  
LIST_PREPROCESS=  
STORE_POSTPROCESS=  
UPDATE_POSTPROCESS=  
GET_POSTPROCESS=  
READ_POSTPROCESS=  
LIST_POSTPROCESS=  
REPLACE_POSTPROCESS=
```

Please note: See also “Shared Application Variable Settings” on page 5-30.

Configuring the STEP AP203 Translator

The STEP Interface recognizes the STEP AP203 Translator (for CADD5), if available on your system. See the *STEP Translators User Guide* for the Export options.

Please note: The STEP AP203 Translator is purchased and installed separately.

The following settings may be useful when using this translator. The translator options are specified in the following two files:

```
$EPD_HOME/data/reposit/putap203.ini  
$EPD_HOME/data/reposit/getap203.ini
```

To change the options, copy these files to your local directory and edit the values.

The following table describes the parameters and default values of `getap203.ini`, which specifies options for the STEP AP203 Translator. For more information, refer to the *STEP Translators User Guide*.

Parameter Name	Type	Description and Default Value
log_file	string	Specifies the output log filename. The default is <code>getap.log</code> .
max_msg	int	Specifies the maximum number of times the same message (error or warning) is written to the log file. The default is -1. A value of -1 specifies no limit.
part_precision	string	Double — Create the CADDs part with double precision. This is the default. Single — Create the CADDs part with single precision.
split	string	No — Do not perform continuity checking on curves and surfaces and do not smooth or split the entity. This is the default. Yes — Perform continuity checking on curves and surfaces and smooth or split the entity when appropriate.
smooth	string	No — Do not smooth. This is the default. Yes — Smooth surfaces that have internal discontinuities. The surface shape may change as a result of smoothing.
cont_tol	float	Sets the linear distance tolerance to detect discontinuities. The default value is 0.0001.
ang_tol	float	Sets the angular tolerance to detect discontinuities. The default value is 0.5 degree.
pnt_tol	float	Defines a general tolerance to determine if two points are the same. This value is used in all tolerance calculations except when checking continuity. The default value is 0.001.

The following table describes the parameters and default values of `putap203.ini`, which specifies options for the STEP AP203 Translator. For more information, refer to the *STEP Translators User Guide*.

Parameter Name	Type	Description and Default Value
log_file	string	Output log filename. The default is <code>putap.log</code> .
max_msg	int	Specifies the maximum number of times the same message (error or warning) is written to the log file. The default is -1. A value of -1 specifies no limit.

Parameter Name	Type	Description and Default Value
class_filter	string	Specifies the AP203 class, which determines the geometric data that is to be filtered from CADDs. This will leave just the geometry of the desired class. All — Translates all data. This is the default. C2 — Translates surface and wire frame geometry only. C4 — Translates surface with topology only. C6 — Translates brep with topology only.
class_force	string	Specifies the highest class of geometry that will be created. All geometry of higher classes will be forced to the class specified. All — Outputs geometry of all classes. This is the default. C4 — Converts solids C6 to C4 class. Any C2 class geometry will be output as C2. C2 — Converts all entities to surface and wire frame geometry only (no topology).
pnt_tol	float	Specifies the point coincidence tolerance. The default value is 0.00001.
subf_explode	string	No — Do not explode subfigures. This is the default. Yes — Explode subfigures. Translate constituent geometry to the parent shape.
description	string	String value used in the STEP file header section. It describes the contents of the STEP file. The default value is None.
author	string	String value used in the STEP file header section. It specifies the author of the STEP file. The default value is None.
organization	string	String value used in the STEP file header section. It specifies the organization of the author. The default value is None.
authorization	string	String value used in the STEP file header section. It specifies the person who authorized this STEP file. The default value is None.

Local Use of EPD Interfaces

To use the EPD Interface for STEP AP203 with EPD.Connect locally, disable the existing Vault support. For more information, refer to the section “Connecting to the Vault” on page 2-10.

If you have not disabled the Vault support, and you want EPD.Connect to list and recognize the application interface types, the first entry in the `pm.config` must be for a Vault that you have configured for the installed interfaces.

Shared Application Variable Settings

The following variables apply to the following interfaces:

- CADDs
- CATIA
- MEDUSA
- Pro/ENGINEER
- STEP AP203

Please note: Generic behavior of the EPD.Connect interfaces is controlled by the following variables defined in the Connect repository file `EPD_HOME/data/reposit/epdconn.ini`.

epdconn.ini

The following settings from the `epdconn.ini` file are used by the EPD.Connect interfaces.

Setting	Description
<code>DO_REVISION_LINKING: Y/N</code>	<p>This should be set to N to use the command line interface.</p> <p>If set to Y, the rulebase creates the links for the parts created in the local area when you do a <code>GET/READ</code> from the vault.</p> <p>If you do a <code>GET</code> for a CADDs part called <code>test</code> with rev 1, you get a copy of the part <code>test</code> in your local area and a link is created from <code>test-1</code> to <code>test</code>.</p> <p>If you do a <code>READ</code> for CADDs part called <code>test</code> with rev 1, a part called <code>test-1</code> is created in your local area.</p> <p>This is applicable only on UNIX platforms.</p>
<code>REMOVE_STORED_FILES: Y/N</code>	<p>This should be set to Y in case you use the Command Line Interface.</p> <p>If set to Y, the rulebase removes the parts from your local area that are transferred to the Vault.</p>
<code>CREATE_PVS: N</code>	<p>This should be set to N. Do not change this setting.</p>
<code>TOOLKIT_TRACE: OFF</code>	<p>This should be set to OFF.</p>

Setting	Description
EXTRACT_PSATTRS: Y/N	This should be set to Y. It is used for enabling the extraction of product structure attributes.
CADDSRB_REPOSIT: \$EPD_HOME/data/reposit/ca ddsrb.ini	Default location of the CADDS interface repository file.
PTCRB_REPOSIT: \$EPD_HOME/data/reposit/pt carb.ini	Default location of the Pro/ENGINEER interface repository file.
MEDUSARB_REPOSIT: \$EPD_HOME/data/reposit/me dusarb.ini	Default location of the MEDUSA interface repository file.
CATIARB_REPOSIT: \$EPD_HOME/data/reposit/ca tiarb.ini	Default location of the CATIA interface repository file.
STEPRB_REPOSIT: \$EPD_HOME/data/reposit/st eprb.ini	Default location of the STEP interface repository file.
CVPATH: \$HOME/parts	The location of the parts directory for the CADDS interface.
STEP_INSTALLED: true/false	Default - false Set to true if the STEP interface is installed
CVCREATESTEP: \$HOME/step	The location of the parts directory for the STEP Interface.
CA_READPREFIX: \$HOME/parts/read	The location of the READ directory. The parts are copied in this directory for READ commands.
CVCREATEDIR: \$HOME/parts	The location of the CREATE directory for CADDS Parts. The parts are copied in this directory for GET command.
CA_ASSYPREFIX: \$HOME/parts	The location of the CREATE directory for PS/CAMU files. The parts are copied in this directory for GET command. This is valid for EPD.Connect interfaces for Pro/ENGINEER, CATIA, MEDUSA.
EPD_HOME: /opt/epd/dm/v60	The location of the Optegra directory.
AWLANG: C	The setting for the language.
CA_BINDIR: \$EPD_HOME/bin	The bin directory for the Optegra software.
EDM_DM_USER: pdmdm	User ID for accessing EDM tables.

Setting	Description
EDM_DM_PASSWD: <encrypted password>	User password for accessing EDM tables.
RB_EXTRACT: \$CA_DATA/appl	<p>The location of the Application specific extract scripts directory.</p> <p>The default location: \$CA_DATA/appl</p> <p>If you have customized extract scripts, set this variable to the appropriate directory. This directory must follow the same directory structure as available in \$EPD_HOME/data/appl.</p>
RB_EXTRACT_PS_DIR:\$RB_EXTRACT/PS	<p>The location of the directory where the extract script for application PS resides.</p> <p>The default location: \$RB_EXTRACT/PS</p> <p>For example: To use the customized extract scripts for PS, set RB_EXTRACT to /usr1/custom dir. The extract script should be placed in /usr1/custom/PS/scripts directory.</p> <p>Please Note: The RB_EXTRACT_PS_DIR setting has precedence over the RB_EXTRACT setting. This setting is applicable only for PS.</p>
RB_EXTRACT_ASSY_DIR:\$RB_EXTRACT/CAMU	<p>The location of the directory where the extract script for application CAMU resides.</p> <p>The default location: \$RB_EXTRACT/CAMU</p> <p>If you have customized the extract scripts, set this variable to the appropriate directory. This directory must follow the same directory structure as available in \$EPD_HOME/data/appl/CAMU</p> <p>For example; To use the customized extract scripts for CAMU, set RB_EXTRACT_ASSY_DIR to /usr1/custom/CAMU directory. The extract script should be placed in /usr1/custom/CAMU/scripts directory.</p> <p>Please Note: The RB_EXTRACT_ASSY_DIR setting has precedence over the RB_EXTRACT setting. This setting is applicable only for CAMU.</p>

The first five variables control the generic behavior of the EPD.Connect interfaces. If defined in the `epdconn.ini` or `cvepd.ini` files, these settings become common for all EPD.Connect interfaces. In the event that you need to reset these variables, move them to interface-specific `.ini` files and then set for the individual interfaces.

The following variables specify repository files for the EPD.Connect interfaces for CADDs, Pro/ENGINEER, MEDUSA, STEP, and CATIA applications.

Variable	Application
CADDSRB_REPOSIT=\$EPD_HOME/data/reposit/caddsrb.ini	CADDs
PTCRB_REPOSIT=\$EPD_HOME/data/reposit/ptcarb.ini	Pro/ENGINEER
MEDUSARB_REPOSIT=\$EPD_HOME/data/reposit/medusarb.ini	MEDUSA
CATIARB_REPOSIT=\$EPD_HOME/data/reposit/catiarb.ini	CATIA
STEPRB_REPOSIT=\$EPD_HOME/data/reposit/steprb.ini	STEP

Repository Files for EPD.Connect Interfaces

This information applies to the following interfaces for EPD.Connect:

- CADDs
- Pro/ENGINEER
- MEDUSA
- CATIA
- STEP

The behavior of all the preceding interfaces is controlled by the settings in three repository files. These files are listed next in order of their precedence:

1. `cvepd.ini`
2. `epdconn.ini`
3. Interface-specific `.ini` file (Each interface has its own specific `.ini` file)
 - `caddsrb.ini`—CADDs
 - `ptcarb.ini`—Pro/ENGINEER
 - `medusarb.ini`—MEDUSA
 - `catiarb.ini`—CATIA
 - `steprb.ini`—STEP

cvepd.ini

Use the environment variable `CVEPD_INIT` to specify the location for the `cvepd.ini` file. This file can be customized to override any settings within

the `epdconn.ini` file or the interface-specific `.ini` file. The settings in this file have higher precedence over the other two repository files.

This file is not mandatory.

Please note: If you are using the Command Line Interface and you do not have a customized `cvepd.ini` file, set `CVEPD_INIT` to `"`. This eliminates the warning `"Env variable CVEPD_INIT is not set !!"`

epdconn.ini

Use the environment variable `CVEPD_MAIN_CONFIG` to specify the location for the `epdconn.ini` file. Set `CVEPD_MAIN_CONFIG` to its default location, `$EPD_HOME/data/reposit/epdconn.ini`. If you have a customized version of this file, set `CVEPD_MAIN_CONFIG` accordingly.

The settings in this file have a higher precedence over the interface-specific `.ini` file.

Interface-Specific .ini Files

caddsrp.ini

Use the environment variable `CADDSRB_REPOSIT` to specify the location for the `caddsrp.ini` file. In the `epdconn.ini` file, this is set to its default location, `$EPD_HOME/data/reposit/caddsrp.ini`. If you have a customized version of this file, set `CADDSRB_REPOSIT` accordingly.

You can override settings for the variables specified in the `caddsrp.ini` file by specifying them in the `cvepd.ini` file.

ptcarb.ini

Use the environment variable `PTCRB_REPOSIT` to specify the location for the `ptcarb.ini` file. In the `epdconn.ini` file, this is set to its default location, `$EPD_HOME/data/reposit/ptcarb.ini`. If you have a customized version of this file, set `PTCRB_REPOSIT` accordingly.

You can override settings for the variables specified in the `ptcarb.ini` file by specifying them in the `cvepd.ini` file.

medusarb.ini

Use the environment variable `MEDUSA_REPOSIT` to specify the location for the `medusarb.ini` file. In the `epdconn.ini` file, this is set to its default location, `$(EPD_HOME)/data/reposit/medusarb.ini`. If you have a customized version of this file, set `MEDUSA_REPOSIT` accordingly.

You can override settings for the variables specified in the `medusarb.ini` file by specifying them in the `cvepd.ini` file.

catiARB.ini

Use the environment variable `CATIA_REPOSIT` to specify the location for the `catiARB.ini` file. In the `epdconn.ini` file, this is set to its default location, `$(EPD_HOME)/data/reposit/catiARB.ini`. If you have a customized version of this file, set `CATIA_REPOSIT` accordingly.

You can override settings for the variables specified in the `catiARB.ini` file by specifying them in the `cvepd.ini` file.

steprb.ini

Use the environment variable `STEPRB_REPOSIT` to specify the location for the `steprb.ini` file. In the `epdconn.ini` file, this is set to its default location, `$(EPD_HOME)/data/reposit/steprb.ini`. If you have a customized version of this file, set `STEPRB_REPOSIT` accordingly.

You can override settings for the variables specified in the `steprb.ini` file by specifying them in the `cvepd.ini` file.

Order of Precedence for Variables Used by the EPD.Connect Interfaces

1. Variables set in the environment
2. Variables set in the `cvepd.ini` file
3. Variables set in the `epdconn.ini` file
4. Variables set in the interface-specific `.ini` file

Preprocess and Postprocess Variables for the EPD.Connect Interfaces

All the interface specific `.ini` files contain the following variables that can be set to the user-developed executable binary or script. The EPD Interface supplies two arguments to these executables: **input filename** and **output filename**. For more information, refer to the *EPD.Connect User Guide*, Appendix C, “Process Flow for the EPD Interface Commands.”

During the installation, these variables are set to NULL:

```
STORE_PREPROCESS=  
UPDATE_PREPROCESS=  
GET_PREPROCESS=  
READ_PREPROCESS=  
REPLACE_PREPROCESS=  
LIST_PREPROCESS=  
STORE_POSTPROCESS=  
UPDATE_POSTPROCESS=  
GET_POSTPROCESS=  
READ_POSTPROCESS=  
REPLACE_POSTPROCESS=  
LIST_POSTPROCESS=
```

Configuring EPD.Connect for Vault

This chapter describes how to configure EPD.Connect for Vault. You can purchase Vault separately.

- Setting the ANSPATH Variable
- Using CADDs CAMU with Vault
- Setting the Maximum Number of Users Within EPD.Connect
- Multiple Vault Access: the pm.config File
- Accessing Vault from a Different Domain
- Running Two Vaults on the Same Server
- Setting Up an Oracle/Vault Account
- Launching and Starting Vault
- Setting Up the Vault Interaction Panels

Setting the ANSPATH Variable

The `ANSPATH` environment variable points to the `pm.config` file, which defines the Vault or Vaults to which a client has access. “Multiple Vault Access: the `pm.config` File” on page 6-3 discusses the file’s contents and format.

UNIX

On the UNIX operating system, to run Vault from within EPD.Connect, you must have Vault loaded and set the `ANSPATH` environment variable in your environment, as in the following example:

```
ANSPATH $HOME/pm.config.
```

Windows

For Windows, you must set the variable in either your environment or `cvepd.ini`. The variable must contain the full path address of the `pm.config` file, as in the following example:

```
ANSPATH=c:\epd\dm\current_revision\data\pm.config
```

Using CADDSS CAMU with Vault

See “Required Scripts for Vault and CADDSS” on page 4-7, “Installing EPD.Connect with CADDSS CAMU on the Same System as Vault” on page 4-7, and “Installing EPD.Connect with CADDSS on a System with Remote Access to Vault” on page 4-17.

Setting the Maximum Number of Users Within EPD.Connect

An Application Entity (AE), called `EDMOSRV_Client`, is defined in the Vault `nsm.config` file. This AE regulates the maximum number of users on the system at one time (concurrent users) to improve efficiency. `EDMOSRV_Client` provides Structured Query Language (SQL) support to EPD.Connect through Oracle.

You must enter the maximum concurrent use of EPD.Connect per node into the `nsm.config` file. This will be reflected in the `MAXINST` setting of the `EDMOSRV_Client` AE. The default value is 15 concurrent connections per node.

Example:

```
# AE for EDMOSRV Client/EPD.Connect
AE(EDMOSRV_Client)
CLOSE
MAXINST(15)
```

For all client nodes, the standard client model has been enhanced to include the previous definition. Client node changes should be made in the model if necessary.

Multiple Vault Access: the pm.config File

The `pm.config` file defines the Vault or Vaults to which a client has access. The file consists of one or more `RESOURCE` statements, entered as continuous lines. Each `RESOURCE` statement identifies one vault to which the client can have access.

The following is an example of a `pm.config` file:

```
RESOURCE(MATTHEW:::matthew:process_manager_domain:
process_manager_AE:0,udp,9000)

RESOURCE(APHIDS:::aphids:process_manager_domain:
process_manager_AE:0,udp,9000)

RESOURCE(RAGTAG:::ragtag:process_manager_domain:
process_manager_AE:0,udp,9000)
```

The following example shows the `RESOURCE` statement syntax expanded to enable the correlation of a Vault name to a Process Manager's AE address:

- The first parameter, separated by three colons (for example, `MATTHEW:::`) represents the `RESOURCE` name. This name represents an Optegra DOMAIN as it is configured in the `nsm.config` file.

- The remaining portion of the `RESOURCE` statement is identical to previous releases of EDM Vault, now known as Vault.

Accessing Vault from a Different Domain

If you are running Vault and EPD.Connect from different network domains, then you must specify the full name of the node in the `pm.config` file, that is, the `ANSPATH` to resolve the full node name.

The following is an example of a `pm.config` file in this case:

```
RESOURCE (APHIDS::aphids:process_manager_domain:  
process_manager_AE:0,udp,aphids.mass.xyz.com,9000)
```

Where, `aphids.mass.xyz.com` is the full node name.

Also, set the value of the environment variable `EDMOANS` to 1.

On the UNIX platform, you can set the value of the environment variable as follows:

```
setenv EDMOANS 1
```

On the Windows NT platform, you can set the value of the `EDMOANS` variable as follows:

Start>Settings>Control Panel>System>Environment

Running Two Vaults on the Same Server

You can run two Vault revisions on the same server by setting the UDP port number in the `nsm.config` and `pm.config` files. Each Vault must have a unique UDP port and `EDMOPORT` variable. You must modify the `pm.config` file on all Vault clients and the `nsm.config` file on the Vault server.

Set the environmental variable (`EDMOPORT`) to a different port number (for example `EDMOPORT=2556;export EDMOPORT`) in the `edmosrv` startup script on the server. Set the same variable on the corresponding clients.

Failure to set the EDMOPORT variable correctly before connecting to the Vault routes EPD.Connect queries to the wrong server.

Setting Up an Oracle/Vault Account

Set up an Oracle account using variables found in the `epdconn.ini` file:

Variable	Description	Value
EDM_QF_USER	Oracle account name of the IQF user	pdmqf
EDM_QF_PASSWD	Oracle account password of the IQF user	Encrypted password using scramexe utility
EDM_DM_USER	Oracle account name of the Vault database owner	pdm dm
EDM_DM_PASSWD	Oracle account password of the Vault database owner	Encrypted password using scramexe utility

Launching and Starting Vault

You can use variables to specify an individual user's sign-on name and password and automatically sign on a user to Vault. To enable automatic sign-on, you must also specify a password.

Use variables found in the `epdconn.ini` file.

Defining the name and password enables EPD.Connect to enter these as defaults in the Sign-on window. The user name defaults to the user's system login name. The variables you can use for general Vault setups are described in the following table.

Variable	Description	Value
CA_EDM_USER	Specifies the default Vault user name.	\$USER (default)
CA_EDM_PASSWORD	Specifies the Vault password.	
CA_AUTO_SIGNON	Specifies whether or not automatic sign-on to Vault should occur at startup.	No (default) Yes
CA_AUTO_LOAD	Specifies whether or not to load a specified tree at startup.	No (default) Yes

Setting Up the Vault Interaction Panels

You can use variables to set default values and change the labels on some fields in the Vault Store panel, the Vault Transfers panel, and the Vault Query Options panel. These variables are described in the following sections.

Setting Up the Vault Store Panel

The Vault Store panel opens when you select Transfers > Store from the EPD.Connect menu bar.

Use variables found in the `epdconn.ini` file. The variables that set default values in the Vault Store panel are described in the following table:

Table 6-1 Environment Variables for Setting Defaults in the Vault Store Panel

Variable	Description	Value
CA_STORE_SIGNOUT	Specifies the default value of the signed-out toggle.	1 — Yes 2 — No (default)
CA_STORE_CLASS	Specifies the default classification for a stored file.	1 — Public 2 — Private 3 — Project (default) - the default setting for the project name is null.
CA_STORE_PROJECT	Specifies the default project for a stored file.	Null (default)
CA_UTYPE_LABEL	Overrides the default label for the User Type field (in both the Vault Store and Vault Query Options panels).	User Type
CA_UTYPE_DEFAULT	Specifies the default value for the User Type field	Null (default)
CA_UTYPE_READONLY	Sets the User Type field to read-only.	0 — Write enabled (default) 1 — Read-only
CA_STYPE_LABEL	Overrides the default label for the System Type field (in both the Vault Store and Vault Query Options panels).	System Type
CA_STYPE_DEFAULT	Specifies the default value for the System Type field.	Null (default)

Table 6-1 Environment Variables for Setting Defaults in the Vault Store Panel

Variable	Description	Value
CA_STYPE_READONLY	Sets the System Type field to read-only.	0 — Write enabled (default) 1 — Read-only
CA_PARTNO_LABEL	Overrides the default label for the Part Number field (in both the Vault Store and Vault Query Options panels).	Part No
CA_PARTNO_DEFAULT	Specifies the default value for the Part Number field.	Null (default)
CA_PARTNO_READONLY	Sets the Part Number field to read-only.	0 — Write enabled (default) 1 — Read-only
CA_DESCR_LABEL	Overrides the default label for the Description field (in both the Vault Store and Vault Query Options panels).	Description
CA_DESCR_DEFAULT	Specifies the default value for the Description field.	Null (default)
CA_DESCR_READONLY	Sets the Description field to read-only.	0 — Write enabled (default) 1 — Read-only
CA_GTCODE_LABEL	Overrides the default label for the GT Code field (in both the Vault Store and Vault Query Options panels).	GT Code
CA_GTCODE_DEFAULT	Specifies the default value for the GT Code field.	Null (default)
CA_GTCODE_READONLY	Sets the GT Code field to read-only.	0 — Write enabled (default) 1 — Read-only

Setting Up the Vault Transfers Panel

The Vault Transfers panel opens when you select Database > List/Transfer from the EPD.Connect menu bar.

Use variables found in the `epdconn.ini` file. The variables that set default values in the Vault Transfers panel are described in the following table.

Table 6-2 Environment Variables for Setting Defaults in the Vault Transfers Panel

Variable	Description	Value
CA_LT_APPLICATION	Default application type.	CADDS
CA_LT_LOCAL_NAME	Local file name to be searched for.	* (wildcard)
CA_LT_REVISION	Default revision.	Null (default)
CA_LT_LOCATION	Default location of the configuration.	1 — Local Disk (default) 2 — Database
CA_LT_DB_NAME	Default name to be searched for in the database.	Null (default)

Setting Up the Vault Query Options Panel

The Vault Query Options panel opens when you select Options > EDM Options from the Vault Transfers panel.

Use variables found in the `epdconn.ini` file. The variables that set default values in the Vault Query Options panel are described in the following table.

Table 6-3 Environment Variables for Setting Defaults in the Vault Query Options Panel

Variable	Description	Value
CA_LT_CLASS	Default classification.	1 — Any (default) 2 — Public 3 — Private 4 — Project
CA_LT_PROJECT	Default project.	Null (default)
CA_LT_STATUS	Default value for the Status field.	Null (default)
CA_LT_USEDBY	The user ID of the person using the item.	user ID
CA_LT_NODE	Default host name.	Null (default)

Table 6-3 Environment Variables for Setting Defaults in the Vault Query Options Panel

Variable	Description	Value
CA_LT_SYSCODE	Default system code.	1 — None (default) 2 — Archive 3 — Delete 4 — Reserve 5 — Review 6 — In-use
CA_LT_USERTYPE	Default value for the User Type field.	Null (default)
CA_LT_SYSTYPE	Default value for the System Type field.	Null (default)
CA_LT_PARTNO	Default value for the Part Number field.	Null (default)
CA_LT_DESCR	Default value for the Description field.	Null (default)
CA_LT_GTCODE	Default value for the GT Code field.	Null (default)
CA_LT_UPDATE	The user ID of the person who last updated the item.	user ID
CA_LT_UPBSO	When the update operation occurred relative to the date.	0 — No entry 1 — Before 2 — Since 3 — On
CA_LT_UPDATE_DATE	Date on which the item was last updated.	Date in YYMMDD format
CA_LT_STORE	The user ID of the person who last stored the item.	user ID
CA_LT_STOREBSO	When the store operation occurred relative to the date.	0 — No entry 1 — Before 2 — Since 3 — On
CA_LT_STORE_DATE	Date on which the item was last stored.	Date in YYMMDD format
CA_LT_ACTION	The user ID of the person who last performed a Vault operation affecting the item.	user ID
CA_LT_ACTIONBSO	When the operation occurred relative to the date.	0 — No entry 1 — Before 2 — Since 3 — On
CA_LT_ACTION_DATE	Date on which the last Vault operation affecting this item occurred.	Date in YYMMDD format
CA_LT_ATTRSET	Attribute set.	Null (default)
CA_LT_ATTRIBUTE	Attribute name.	Null (default)
CA_LT_ATTRVALUE	Specifies the attribute value.	Null (default)

Displaying the Labels Used to Describe Vault Items

A revised label appears instead of the standard label on any panel with an attribute field. For example, revised labels appear in the following panels:

- Store panel
- Vault Transfers panel (Database option selected)
- Change File Attributes panel

Attribute descriptions of Vault items change only when displayed in EPD.Connect. They are not changed in the Vault database.

Upgrading Navigator to EPD.Connect

You can optionally upgrade your Navigator product to EPD.Connect as described in this chapter. The following topics are presented:

- Preparing to Upgrade
- Using the Navigator Refresh Tool

Preparing to Upgrade

When preparing to upgrade from Navigator to EPD.Connect, be aware of the following:

- As a precaution, you should back up your Navigator software and your operating system before you upgrade to EPD.Connect.
- You need a previously installed version of Navigator (the executable file from which you are upgrading), the user ID of your Oracle administration account, and the user ID of your Vault administration account.
- You should schedule the upgrade and notify users when it will take place. Suspend Navigator access until the upgrade is complete.

Exiting the Upgrade

You can exit the upgrade tool by pressing `Control-c` to abort the refresh procedure at any time.

If you exit the refresh tool, you can restart it by reexecuting the `$EPD_HOME/install/navrefresh` command file. The tool resumes the upgrade from the beginning, overriding previously entered information.

Using the Navigator Refresh Tool

This section describes how to use the `navrefresh` script to upgrade from Navigator to EPD.Connect.

Starting the Upgrade

You must be logged in as root to run `navrefresh` on the Vault system host.

1. Change to the install directory:

```
# cd $EPD_HOME/install
```

2. Invoke the automated refresh tool. Enter:

```
# navrefresh [| tee navrefresh.log]
```

where,

```
| tee navrefresh.log creates a history log file.
```

The navrefresh nameplate appears:

```
*****  
Database Refresh Utility for EPD.Connect  
*****
```

Accepting Defaults

Default values are shown in brackets ([]). To accept a default, press Return.

```
Enter directory where Vault is installed  
[/opt/epd/dm/v60]:
```

Overriding Defaults

You can override defaults and specify your own value by answering *yes* to this prompt.

```
Do you want to re-enter any values [no] ? yes
```

After you answer *yes*, type a substitute value and press Return.

Entering Account and Path Information

After you invoke `navrefresh`, you are prompted for the following information:

```
Enter the user id of your Oracle Administration  
account: oracle  
Enter the user id of your Vault Administration  
account: edm  
Enter directory where Vault is installed  
[/opt/epd/dm/v60m]: /opt/epd/dm/v60  
Enter directory where Navigator is installed  
[/opt/epd/dm/v60]: /opt/epd/dm/v60
```

Check the input values:

Installation details you have specified are:

```
Oracle Administration account: oracle  
Vault Administration account: edm  
Optegra Vault installation directory: /usr/apl/edm
```

```
Navigator x.x installation directory: /usr/apl/nav
```

The system prompts you to confirm the values or enter new ones.

Updating Oracle Tables

If you choose *yes* when prompted to refresh the Navigator Oracle tables, *navrefresh* enables a GUI revision in Oracle and updates the setting for a local rulebase patch, as in the following dialog between the system and the user:

1. At the following prompt, choose the default (*yes*).

```
OK to refresh the Navigator Oracle Tables [yes]:
```

2. Enter the user name for the Oracle user manager :

```
Enter the Vault Oracle database manager userid [pdmdm]:
```

3. Enter the password for the Oracle user manager:

```
Enter the Vault Oracle database manager password [pdmdm]:  
Updating Tables in ORACLE Database .....
```

Checking for Navigator ADRAW Rulebase Support

If you answer *yes* when prompted for ADRAW rulebase support, *navrefresh* adds a link to the following file:

```
$EPD_HOME/data/EDM.DEFAULTS
```

To check for ADRAW rulebase support, choose the default (*yes*) at the following prompt:

```
OK to check for Navigator ADRAW rulebase support  
[yes]: y  
Complete.
```

When *navrefresh* completes execution, the following is displayed:

```
Navigator Refresh Completed.  
#
```

Configuring EPD.Connect Application Defaults

This chapter describes how to configure application defaults. The following topics are presented:

- Setting Colors
- Setting Component Text
- Setting Fonts for the EPD.Connect Interface
- Scaling the Windows
- Specifying Directories
- Setting Database Defaults
- Setting Up the Open Configuration Window
- Controlling Component Selection
- Controlling Geometric Accuracy
- Controlling the Tessellation File Transfer
- Controlling Clash Detection and Zoning
- Setting Default Application Directories
- Enabling Nested Reference Assemblies
- Setting Information Browser Preferences

You can set conditions for your EPD.Connect application in your local `cvepd.ini` file. The variables described in this chapter reside in the `epdconn.ini` file supplied in `$EPD_HOME/data/reposit`.

Setting Colors

Use variables to set the EPD.Connect colors.

Specifying the Color Palette

You select the EPD.Connect colors using the `-color` command line option. You can set this option to gray (the default) or blue. This instructs EPD.Connect to use the appropriate palette file.

Setting Colors for the Interface

The variables for setting colors take a default color number as their value. The number is an integer in the range of 0 through 15. The format of the variables is as follows:

`CA_xxxx_COLOR integer`

The following table describes the EPD.Connect variables you can use to set the interface colors:

Table 8-1 Environment Variables for Setting the Interface Color

Variable	Description	Value
<code>CA_REF_COLOR</code>	Reference components color	6
<code>CA_NODE_COLOR</code>	Components color	2
<code>CA_NODE_SEL_COLOR</code>	Selected components color	8
<code>CA_NODE_BACK_COLOR</code>	Selected component text color	10
<code>CA_COMPARE_NEW</code>	Text color of new components after a compare operation	1
<code>CA_COMPARE_CHANGED</code>	Text color of changed components after a compare operation	9
<code>CA_COMPARE_SAME</code>	Text color of unchanged components after a compare operation	11
<code>CA_LINK_COLOR</code>	Component link lines color	11
<code>CA_WU_LINK_COLOR</code>	Where-used link line color	1
<code>CA_TRACE_COLOR</code>	Trace-selected link lines color	3

Setting Component Text

Use variables to set the display of text on components and the initial entries in the Text and Delimiter fields of the Component Text Display window. The following table describes these variables.

Table 8-2 Environment Variables for Displaying the Component Text

Variable	Description	Value
CA_COMPONENT_TEXT	Specifies the default string of items.	\$CLASS-NAME_\$INSTANCE
CA_STRUCTURE_TEXT	Specifies the default string of items.	\$CLASS-NAME_\$INSTANCE
CA_COMPONENT_DELIM	Sets the item-delimiting character in the Component Text window.	Specify a single character (default)

Please note: The length of the component on the displayed tree is determined by the length of the displayed component text string.

Setting Fonts for the EPD.Connect Interface

Use variables to set fonts of text displayed by the EPD.Connect interface and the component tree, including the main tree and the overview tree. You can select text fonts of the components from the Tree Font Customizer window.

The CA_FONT variable defines the text font used for all text in the EPD.Connect window. Other variables specify the font for text used in the menu bar and menus, for the scrolling list areas, and so on.

The CA_TREE_FONT and CA_SMALL_FONT variables affect the text on tree components. The available fonts and defaults assigned to these variables depends on the window system under which EPD.Connect is running.

The following table describes the variables you can use to set fonts. The variables reside in the template files `cfgmotif.ini` and `cfgwin98.ini`.

Table 8-3 Environment Variables for Setting EPD.Connect Fonts

Variable	Description	Value
CA_FONT	Font for text used in the entire EPD.Connect window	*helvetica-bold-r-*--12-*ISO8859-1
CA_MENU_FONT	Font for text used in the menu bar and menus	*helvetica-bold-r-*--14-*ISO8859-1

Table 8-3 Environment Variables for Setting EPD.Connect Fonts (Continued)

Variable	Description	Value
CA_LIST_FONT	Font used for the scrolling list areas	*courier-bold-r-normal--12-*ISO85-1
CA_TREE_FONT	Font used for the main tree component	\$CA_LIST_FONT
CA_SMALL_FONT	Font used for the overview tree components	nil2

Scaling the Windows

You can set a scale factor to accommodate display resolution. The following screen resolution files are supplied in `$EPD_HOME/data/reposit`:

- `cfg800.ini`
- `cfg1024.ini`
- `cfg1280.ini`
- `cfg1152.ini`

Specifying Directories

Primary directories and subdirectories are available to you for configuring EPD.Connect. The variables you use to reference these directories are described in this section.

Primary Directories

Primary directories specify the principal locations for all EPD.Connect data files. The following table describes the variables you use to specify primary directories.

Table 8-4 Environment Variables for Specifying Primary Directories

Variable	Description	Value
CA_DIR	EPD.Connect UNIX script directory	<code>\$EPD_HOME/scripts</code>
CA_SCRIPTS	EPD.Connect scripts directory	<code>\$EPD_HOME/scripts</code>
CA_DATA	EPD.Connect data directory	<code>\$EPD_HOME/data</code>
CA_EXTRACT	Directory containing extract, lock, and create UNIX scripts for each application	<code>\$EPD_HOME/data/extract</code>

Table 8-4 Environment Variables for Specifying Primary Directories (Continued)

Variable	Description	Value
CA_MENU	Directory containing data files for standard menus in all windows with menu bars	\$EPD_HOME/data/menus
CA_TEMPLATES	Directory containing templates for Access, Visibility, and Highlight rules	\$EPD_HOME/data/template
CA_TEMP	Directory for temporary files	/usr/tmp

Subdirectories

Subdirectories specify the location of EPD.Connect runtime files. The following table describes the variables you use to specify subdirectories.

Table 8-5 Environment Variables for Specifying Subdirectories

Variable	Description	Value
CA_HELP	Location of help files	\$CA_DATA/help
CA_ACCESS_TEMPDIR	Directory of templates for Access flags	\$CA_TEMPLATES/access
CA_APPLICATIONS	File containing the application environment definitions in effect when no Vault is configured	\$CA_DATA/custappl/appl_def
CA_COMPARE_REPORT	Directory for the Compare Report menu	\$CA_MENU/ca_compr.men
CA_CUSTOM_MENU	Directory containing user window screen definitions for specific user extension	\$CA_DATA/custmenu/\$AWLANG
CA_CUSTOM_POPUPS	Directory containing user window screen definitions for specific user extension	\$CA_DATA/custpops
CA_FONT_DIR	Directory of font list files	\$CA_DATA/fontlist
CA_HIGH_TEMPDIR	Directory of templates for highlight rules	\$CA_TEMPLATES/hiligh
CA_ICONS	Location of icons	\$CA_DATA/icons
CA_QUERY_DIR	Location of IQF menus and queries	\$CA_DATA/iqfmenus
CA_QUERIES	Location of the main IQF query menu	\$CA_QUERY_DIR/iqfmain.men

Table 8-5 Environment Variables for Specifying Subdirectories (Continued)

Variable	Description	Value
CA_REPORT_DIR	Directory containing the EPD.Connect reports	<code>\$CA_DATA/reports</code>
CA_RESOURCES	Location of the master resource files	<code>\$CA_DATA/resource</code>
CA_STATE_DIR	Directory containing saved tree states	<code>\$CA_DATA/state</code>
CA_CREATE_TRIGGER	Tree modification trigger program	<code>\$CA_SCRIPTS/edit_trigger</code>
CA_TRIGGER	Directory containing all triggers defined by users	<code>\$CA_DATA/trigger</code>
CA_VIS_TEMPDIR	Directory of templates for visibility rules	<code>\$CA_TEMPLATES/visible</code>
CA_CLOSE_ICON	Icon displayed when EPD.Connect windows are closed	<code>\$CA_ICONS/cvlogo3d</code>
CA_BUTTON_DIR	Location of button sets	<code>\$CA_DATA/toolbar/\$AWLANG</code>
CA_BUTTON_LDIR	Directory containing the button definitions for the Vault Transfers window	<code>\$CA_DATA/toolbar/\$AWLANG</code>
CA_BUTTON_FILE	Button set to be displayed when EPD.Connect is initially executed	<code>\$CA_BUTTON_DIR/standard.def</code>
CA_BUTTON_LFILE	Default button definition for the Vault Transfers window	<code>\$CA_BUTTON_LDIR/transfer.def</code>

Setting a Mask for Files or Directories

You can set a mask for files or directories processed by the `get` or `read` commands. The following table describes the variable you use to set a mask for a file or a directory.

Table 8-6 Environment Variable for Setting a Mask for Files or Directories

Variable	Description	Value
CA_CREATEMASK	Sets a mask for a file or a directory.	Default — 002 Directories — 777 - 002 = 775 Files — 666 - 002 = 664

Setting Database Defaults

You can determine the extent to which EPD.Connect users are allowed to set their own defaults for Vault transfer actions, such as deferred transfers. The following table lists and describes the variables you can use to set database defaults.

Table 8-7 Environment Variables for Setting Database Defaults

Variable	Description	Value
CA_ENABLE_PS	Specifies whether or not EPD.Connect is able to work with Product Structure files. For example, if you need EPD.Connect to work only with CAMU assemblies (turn it off in this case).	Yes (default) No
CA_GLOBAL_CHANGE_ITEMNAME	Specifies whether a changed item name is reflected in the entire product structure or not.	1 — Reflected in the entire product structure. 0 — Reflected only in the class of the selected node. (default)
CA_EXPLODE_CONFIG	Specifies whether or not to explode a configuration on storing it in the Vault. You can use an exploded configuration to search for components in the Vault database.	No — No Value, 0 Yes — Vault-type EDM5ATTR Set the value to EDM5ATTR to explode a configuration into Vault-type attributes when it is stored in the Vault.
CA_PRESERVE_ATTRS	Preserves Vault user-defined attributes upon an update or replace.	1 — Yes 2 — No (default)

Table 8-7 Environment Variables for Setting Database Defaults (Continued)

Variable	Description	Value
CA_FILE_ENV	Specifies the environment for storing files. Also specifies the file or data format.	CADDS (default) — Stores the file using CGOS file-naming conventions. Local — Maintains local file-naming conventions (UNIX format).
CA_REV_INTERP	Sets the display or nondisplay of revision levels attached to local files while the local database is being queried.	1 — Display local file revision levels in the revisions field (default) 0 — Display the files as they are named on the local disk
CA_READ_OVERWRITE	Controls the overwriting of files in the Read directory. When set to Yes, this allows a file or part previously retrieved with a READ command to be overwritten by a subsequent READ of the same file or part. Note: This variable governs overwriting files in the Read directory.	1 — No 2 — Yes (default)
CA_GET_OVERWRITE	Controls the overwriting of files retrieved with a GET command. When set to Yes, this allows a file or part previously retrieved with a GET command to be overwritten by a subsequent GET of the same file or part.	1 — No (default) 2 — Yes
CA_CHECK_INSTANCE	Sets instance checking when a component is added to a tree. Note: You normally use instance checking to ensure that items on a tree have unique identifiers (for example, in trees that represent product assemblies).	1 — On (default) 2 — Off
CA_EDIT_ON	Controls user access to the Edit menu; disallows editing of a tree open for read-only access.	1 — Allow edits (default) 0 — Disallow edits

Table 8-7 Environment Variables for Setting Database Defaults (Continued)

Variable	Description	Value
CA_EDM_DEFERRED	Enables or disables Vault batch processing.	Yes (default) No
CA_DYNAMIC_PULLDOWN	Enables or disables querying the database each time a selection list is activated. (A selection list offers options for certain attributes for Vault items, for example, Project.) Note: With dynamic querying enabled, the Vault database is queried again each time a selection list is activated. Querying the database automatically updates the selections list. You can usually disable dynamic querying after you have configured the database for a production environment.	Yes — Enables dynamic querying (default) No — Disables dynamic querying
CA_LIST_LOCAL_LONG	Operating system command for a long local listing.	ls -lag
CA_LIST_LOCAL_SHORT	Operating system command for a short local listing.	ls
CA_PRINT_COMMAND	Specifies the operating system print command.	lpr
CA_SAVE_TEMPLATE	Sets the availability of the template SAVE button.	1 — Visible (default) 0 — Not visible
CA_LINK_FOLLOW	Enables the Vault Transfer window to list the local file system and follow any symbolic links.	0 — False (default) 1 — True

Setting Up the Open Configuration Window

You can configure the default settings for the Open Configuration window using the variables described in the following table.

Table 8-8 Environment Variables for Setting Open Configuration Window Defaults

Variable	Description	Value
CA_CONFIGURATION	Specifies the default name in the Open Configuration dialog box.	Null (default)
CA_APPLICATION	Specifies the default type of structure in the Open Configuration dialog box.	PS and CAMU
CA_REVISION	Specifies the revision of the default configuration.	Null (default)
CA_LOCATION	Specifies the default configuration location.	1 — Local disk 2 — Vault database (available only if CA_EXPLODE_CONFIG is set to Vault-typeATTR) (default)
CA_OPEN_DIR_NO	Presets the directory option.	1 — Assembly Create directory (default) 2 — Assembly Read directory 3 — A user-specified directory
CA_ENABLE_LOCKING	Specifies if a lock file is to be created when opening an assembly for Write access. Note: When CA_ENABLE_LOCKING=No in the Open Configuration dialog box, File Mode is not displayed. The only mode available is the Write mode.	Yes (default) No
CA_LOCKING_DEFAULT	Specifies the default setting of the File Mode button in the Open Configuration dialog box.	1 — Read-only (default) 2 — Write

Table 8-8 Environment Variables for Setting Open Configuration Window Defaults (Continued)

Variable	Description	Value
CA_QUERY_INFO	Specifies the level of attribute querying.	0 — No query to be performed 1 — Local disk query only 2 — Vault database query only 3 — Local disk and Vault query (default)
CA_MAX_ITEMS_PER_CACHE	Specifies the maximum number of items cached when the Open Configuration dialog box is invoked.	Any number greater than 100 and less than 10000. Default is 5000.

Controlling Component Selection

The variables described in the following table set the component selection and the initial toggle button settings on the Display Control window.

Table 8-9 Environment Variables for Tree Display

Variable	Description	Value
CA_SELECT_STATE	Specifies SINGLE or MULTI component selection.	1 — SINGLE 2 — MULTI (default)
CA_SELECT_MODE	Specifies NODE or SUBASSEMBLY settings.	1 — NODE (default) 2 — SUBASSEMBLY

Controlling Geometric Accuracy

The variables described in the following table specify the accuracy of the storage and positioning of EPD.Connect information.

Table 8-10 Environment Variables for Controlling Geometric Accuracy

Variable	Description	Value
CA_DISPLAY_PREC	The screen display precision.	4
CA_OUTPUT_PREC	The precision with which to save data to disk.	12
CA_REPORT_PREC	The precision with which to incorporate values into reports.	12

Table 8-10 Environment Variables for Controlling Geometric Accuracy

Variable	Description	Value
CA_ORIENT_XYZ_TOL	The position attribute tolerance.	0.001 (default)
CA_ORIENT_ANG_TOL	The angle attribute tolerance.	0.001 (default)

Controlling the Tessellation File Transfer

The following table describes the `CA_CHECK_TD_SUR` variable, which controls the forced transfer of the CADD5 tessellation file.

Table 8-11 Environment Variable for Controlling CADD5 Tessellation File

Variable	Description	Value
CA_CHECK_TD_SUR	Validates the date stamp of a CADD5 5_t.d file.	A three-digit number specifying the options required for the Store, Update, and Replace menu commands: 0 - Do not check 1 - Check and warn if inconsistent 2 - Do not transfer if inconsistent 000 (default)

See “Generating and Locating Graphic Binary Files” on page 4-2.

Controlling Clash Detection and Zoning

You can optimize the performance of clash detection using the variables described in the following table.

Table 8-12 Environment Variables for Controlling Clash Detection and Zoning

Variable	Description	Value
CLASH_MEMORIZE_EXTENTS	Stores the part extents data and does not access the file for each component.	0 - disable (default) 1 - enable for zoning only 2 - enable for zoning and clash detection

Table 8-12 Environment Variables for Controlling Clash Detection and Zoning (Continued)

Variable	Description	Value
CLASH_CHECK_COMP_READ	Accesses the data files containing tessellation information only once during a clash detection.	0 - disable (default) 1 - enable
CLASH_IGNORE_FILE_ERRORS	Continues clash detection despite missing tessellation files. A list of such missing files is stored in a file with a <code>.miss</code> extension along with the report generated.	0 - disable (default) 1 - enable
CLASH_OPTIMIZE_ZONING	Checks each component against all zones so that a component is read only once during a zoning operation. Use this variable when there are only a few zones.	0 - disable (default) 1 - enable
CLASH_ZONING_NOLAY	Specifies ranges of layers to be ignored in a binary graphic file (<code>.gbf</code>) when a clash detection is performed. By default this variable is disabled. Example: CLASH_ZONING_NOLAY 6-9 2 3	Layer numbers
CLASH_FIND_AGAIN_IF_COM_DIS	Searches the directories defined in the CVPATH if a tessellation file is not found in the original directory.	0 - disable (default) 1 - enable
CLASH_IGNORE_DUPLICATE	Lists only one instance of a clash between two components in the Report Viewer.	0 - disable (default) 1 - enable

Table 8-12 Environment Variables for Controlling Clash Detection and Zoning (Continued)

Variable	Description	Value
CLASH_COMP_MAX_CLASHES_FIRST	Sorts components in the Report Viewer according to the number of clashes detected per component in descending order.	0 - disable (default) 1 - enable

Setting Default Application Directories

EPD.Connect variables define directory structures for related applications, control user access to files, and configure a system to meet individual working styles.

The management of local information can cause problems when a number of users work with the same information sets. Examples of applications include CADDs and CAMU. The UNIX file system sometimes presents difficulties when the following occurs:

- Several users share a common disk of read data.
- Several users access assemblies and CADDs assembly drawings.
- Parts read from the Vault for read-only must be locked by CADDs.
- CAMU rules for working with parts are specified at a revision.

In the EPD.Connect interface, windows that request a file name specification for data provide a choice of the directories to be searched. Query windows provide a choice of the following directories for which you can set defaults:

- Assembly Create directory
- Read Parts directory
- CADDs Create directory

Toggle buttons are usually available for making a selection. An additional button is available to specify a directory path name other than a predefined directory.

- For Java classes — You can set the environment variable `CLASSPATH` to search within multiple directories for Java classes. Set this variable to a list of directories, separated from one another with a colon (:), containing Java classes.

- For a site-specific directory for registry and preferences files — You can set the environment variable `EPD_SITE_DIR` to a site-specific directory for these files. The default is `$EPD_HOME/data`.

The following table describes the Assembly Create, Read Parts, and CADDs Create directories. These are user-defined directories for assemblies and CADDs parts.

Table 8-13 Assembly, Read Parts, and CADDs Directories

Directory	Description
Assembly Create	Contains assemblies and assembly drawings that are transferred from Vault for modification.
Read Parts	Contains all data, including assemblies, accessed from Vault for reading.
CADDs Create	Contains all other data extracted from Vault for modification.

The variables you can use to specify these user-defined directories are described in the following table.

Table 8-14 Environment Variables for Defining Assembly and Parts Directories

Variable	Description	Value
<code>CA_ASSYPREFIX</code>	The assembly directory.	<code>\$HOME/parts</code>
<code>CA_READPREFIX</code>	The Read Parts directory.	<code>\$HOME/parts/read</code>
<code>CA_LOCK_READ</code>	Controls access to data by locking parts for CADDs read. (Yes/No) Note: If this variable is set to Yes, parts that are read-accessed may be locked by CADDs for read. This creates a <code>_pd.LOCKW</code> file.	Yes (default) — Lock the parts read from the Vault
<code>CVPATH</code> (defined in the <code>.caddsrc</code> file)	The directories for CADDs parts and related data. In addition to this variable, EPD.Connect uses the CADDs Create directory for CADDs parts and assemblies. The CADDs Create directory is delineated by “=C:”.	<code>\$HOME/parts</code>

The `CA_ASSYPREFIX`, `CA_READPREFIX` and CADDs Create directories can all be the same. If they are different, you must include them all in the

CVPATH definition, so that CADD5 can locate the files. All new data created by CADD5 or EPD.Connect is written to the CADD5 Create directory.

Please note: When you are working with CADD5, include the CA_ASSYPREFIX and CA_READPREFIX directories in the CVPATH definition.

From EPD.Connect, you can perform the following actions:

- View the current paths assigned to the variables.
- Query the local database to determine where the query looks for create or read parts.
- Change the direction of the local search.

Refer to the *EPD.Connect User Guide* for more information.

Enabling Nested Reference Assemblies

You can enable nested reference assemblies of CAMU in the assembly mode. The following table describes the variable you can set for opening a nested reference assembly.

Table 8-15 Environment Variable for Opening a Nested Reference Assembly

Variable	Description	Value
CAMU_EXPAND_REF_ENABLE	Enables the nested reference assembly.	0 - For CADD5 5 Revisions 8, 9, and 10 1 - For CADD5 5i Release 11

Setting Information Browser Preferences

You can set preferences to the Information Browser. For details refer to the *Information Browser User Guide*.

Customizing EPD.Connect Menus and Reports

This chapter describes how to customize menus, buttons, and reports. The following topics are presented:

- Customizing the Toolbar
- Customizing Menus
- Generating Tree Reports
- Generating Tree Comparison Reports
- The Report Data File

To customize application defaults such as color and font, see Chapter 8, “Configuring EPD.Connect Application Defaults.”

Customizing the Toolbar

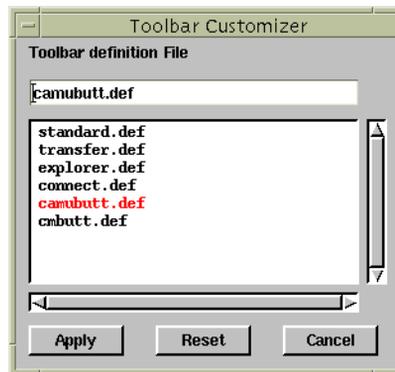
The EPD.Connect toolbar appears at the top of your screen while EPD.Connect is running. The toolbar contains buttons that initiate specific actions. You can assign standard EPD.Connect commands to buttons and display sets of buttons on the toolbar. This allows you to set up the toolbar to quickly choose the commands that you most frequently use.

To display a specific toolbar, follow these steps:

1. Select Window from the EPD.Connect top bar.
2. Select Toolbar Customizer from the menu.

The Toolbar Customizer menu appears.

3. In the Toolbar definition file field, specify an existing toolbar definition file and click Apply. Its contents are displayed in the toolbar menu area.



Default Toolbar File Locations and Pointers

EPD.Connect looks for button definitions and icons in specific directories. These directories normally reside in EPD.Connect's data directory.

The default location of toolbar definition file directories and their files is:

`$EPD_HOME/data/toolbar`

The following table describes the variables specifying these directory and files.

Table 9-1 Environment Variables for Buttons and Icons Directories

Variable	Description	Value
CA_BUTTON_DIR	The directory containing the toolbar button definitions.	
CA_BUTTON_LDIR	The directory containing the button definitions for the Vault Transfers window.	
CA_BUTTON_FILE	The file containing the button set that is displayed when EPD.Connect is initially executed.	\$EPD_HOME/data/toolbar/C/standard.def
CA_BUTTON_LFILE	The file containing the default button definition for the Vault Transfers window.	\$EPD_HOME/data/toolbar/C/transfer.def
CA_ICONS	The directory containing the icons.	\$EPD_HOME/data/icons

The standard.def File

The EPD.Connect buttons directory contains a file called `standard.def`. This file contains the default button set that appears when you initially execute EPD.Connect. This file also lists the available actions that can be applied to buttons, with their corresponding action codes. The entries in the list are commented out, so that no buttons are displayed. You can, however, use the file to create new button definition files. You can display up to 28 buttons on the toolbar.

Creating a Customized Button Set

To create a new toolbar display, edit a copy of the `standards.def` file. Each toolbar button requires an entry containing the following items. Each

parameter in the entry line must be separated by at least one space character:
number action_code button_icon command_prompt

Table 9-2 Button Set Parameters

Parameter	Description
number	The position of the button in the button bar.
action_code	A code for the action to be performed. (Refer to Appendix B, "EPD.Connect Action Codes" for descriptions of the action codes.)
button_icon	Defines an icon to be displayed on the button. Icon files have an .icn extension. You must place all icon files in the \$CA_DATA/icons directory.
command_prompt	The text shown in the Assist field when the cursor moves over the button.

The crosshatch (#) in the first character position begins a comment line. Use the crosshatch (#) to comment out positions to which you assign no button. Replace the crosshatch (#) with a button number or duplicate and edit the lines.

Customizing Menus

EPD.Connect provides menus that you can customize by rearranging or removing the options or by supplying application-specific functions. You can customize menus to provide:

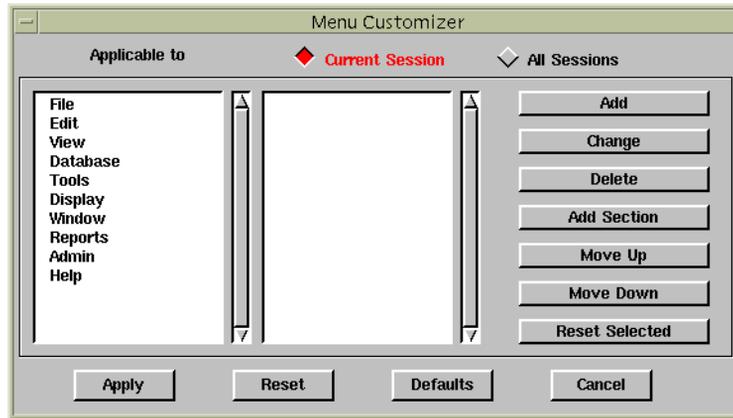
- Specific reporting options
- EPD.Connect command actions
- Commands to send to CADDSS
- Commands to launch other applications

Editing the Contents of a Pulldown Menu

To customize a menu:

1. Select Window from the EPD.Connect top bar.
2. Select Menu Customizer from the menu.

The Menu Customizer appears.



3. The left field displays top bar menu options. Select the one that you want to customize. In this example, File was selected.
4. Click one of the five buttons located at the far right of the menu. Each button allows you to perform that action.

Scrolling Up and Down in a Menu List

To scroll up and down in a menu list:

1. Select an option from either the top bar headings list column or the pulldown options list.
2. Click either the Up or the Down button.

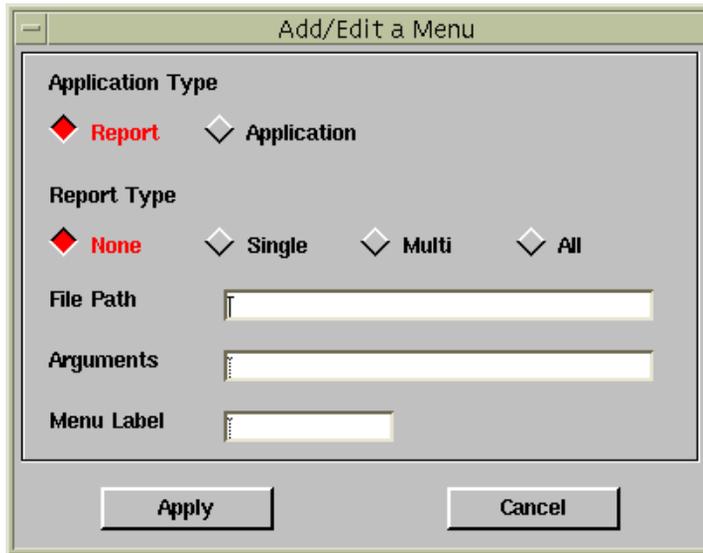
Cancelling Menu Customization

You can close the Menu Customizer by clicking the Cancel button.

Adding a Menu Option

Click the Add button on the Menu Customizer to add an option to any of the top bar pulldown menus.

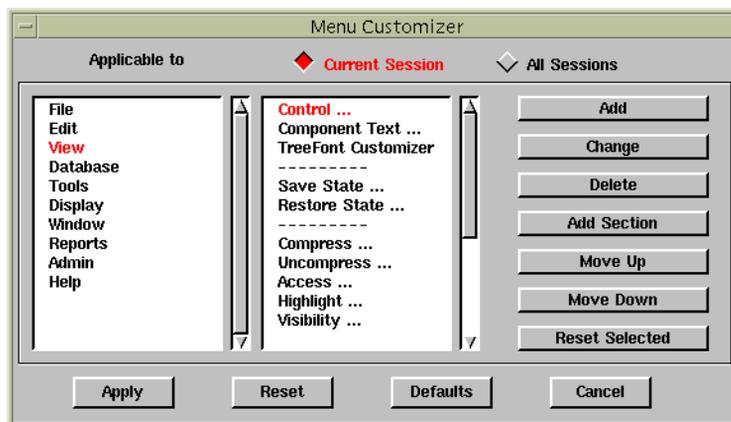
The Add/Edit a Menu property sheet appears.



Deleting a Menu Option

To delete a menu option:

1. Select the menu option you want to delete, for example, Control on the View menu list.



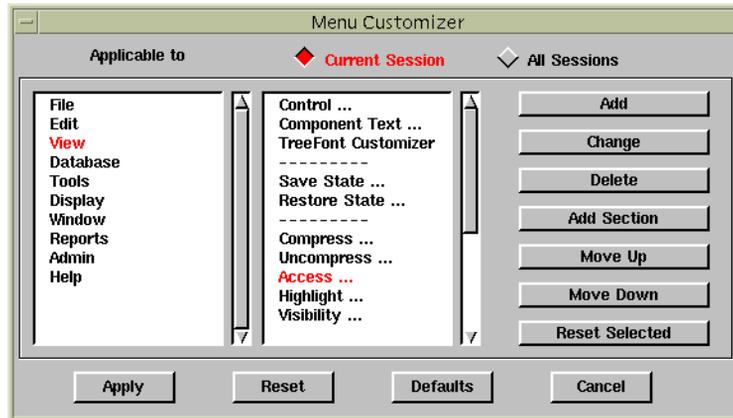
2. Click the Delete button on the Menu Customizer.

The menu option is removed.

Editing a Menu Option

To edit a menu:

1. Select the menu option you want to edit, for example, Access on the View menu.

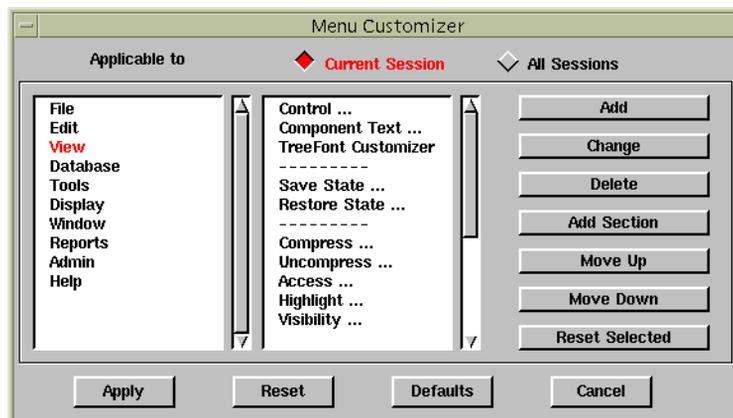


2. Click the Change button on the Menu Customizer.

Sectioning a Menu

By sectioning you can add a section or menu separation line to the pulldown menu. To section a menu:

1. Select the menu option to which you want to add a section.
2. Click the Add Section button on the Menu Customizer.



Creating a Custom Menu Command

A custom menu command specifies the actions that produce a customized report.

The format of an action code 6500 custom menu command is as follows:

```
display_text action_code keyword1 keyword2 command
```

Table 9-3 Custom Menu Command Parameters

Parameter	Description
display_text	Defines the text displayed in the menu.
action_code	Action code 6500 is the EPD.Connect internal command to execute a user-defined script. The action of a custom menu option (which is associated with the action code 6500) is to output an ASCII file defining a set of components governed by keyword2. The script can use this output file to generate an input file to produce a result determined by keyword1.
keyword1	Defines the action to be performed with the report results. These actions are described in the table on page 9-9.
keyword2	Defines the scope for the components to be output. These are described in the table on page 9-9.
command	Defines the scope for the components to be output. These are described in the table on page 9-9.

The format of an action code 6501 custom menu command is as follows:

```
display_text 6501 report#
```

Please note: To add any custom menus, use any integer report number above 3000. Currently, reports number up to 899 are reserved by EPD.Connect (that is, 1-199 for Visualizer Integration, 200-399 for NAV-CAMU Integration, 400-599 for CM Integration, 800-899 for Step interface). EPD.Connect needs to reserve report numbers up to 3000 for

future releases. These report numbers will not be modified for future releases of EPD.Connect.

Table 9-4 Action Code 6501 Custom Menu Command Parameters

Parameter	Description
display_text	Defines the text displayed in the menu.
action_code	Instructs EPD.Connect to use a lookup table to obtain the information required to execute a script.
report#	Instructs EPD.Connect to execute the report number found in the first column of \$EPD_HOME/data/reports/lookup.rep.

Please note: The menu fields are tab separated, not space separated.

Custom Menu Command Keywords

The actions associated with keyword1 and keyword2 are defined in the following tables.

Table 9-5 Keyword1 Descriptions

Keyword1	Description
REPORT	Produces a report to display as a listing. The Report Viewer window opens to display it.
HIGHLIGHT	Highlights a list of components in the current tree. A list of components is read and highlighting applied.
SELECTED	Selects a list of components in the current tree. A list of components is read and the selected state applied.
CADDS	Sends a resultant report file to CADDS. The file returned is issued as CADDS commands.
APPLICATION	Runs an external program, the results of which are not displayed.
PERL	Runs a PERL script.

Table 9-6 Keyword2 Descriptions

Keyword2	Description
NONE	Does not output any components from the tree
SINGLE	Outputs the last selected component from the tree

Table 9-6 Keyword2 Descriptions

Keyword2	Description
MULTI	Outputs all the selected components from the tree
ALL	Outputs all components from the tree

Example of a Custom Menu Command

An example of a custom menu command is as follows:

```
Parts_list 6500 REPORT ALL $CA_REPORT_DIR/report1
```

In this example,

- `Parts_list` is the text displayed in the Reports menu.
- The action code, `6500`, specifies the default action for reports.
- `REPORT` and `ALL` keywords specify that a report is to be generated of all components in the tree.
- `$CA_REPORT_DIR/report1` specifies the command you want executed.

Generating Tree Reports

The custom menu options that create reports use customizable report generation facilities in EPD.Connect. These allow programs to access the data in a displayed tree. The report result depends on the keywords you use when launching the custom report from EPD.Connect. (See page 9-9 for descriptions of keywords.) The report result may be one of the following:

- A text list
- An action applied to the tree (such as highlighting or selecting tree components)
- The loading of data into CADDs or another application

Refer to “Customizing the Toolbar” on page 9-2 for information about the custom menu options for creating reports.

Tree-Report Directory

EPD.Connect looks for reports in its reports directory. You specify this directory with the `CA_REPORT_DIR` variable. The default for the reports directory is the following:

```
CA_REPORT_DIR:$CA_DATA/reports
```

When you request that EPD.Connect perform one of the custom options from a menu, it performs the following actions associated with its action code 6500:

1. Writes out the data for the specified tree components to an ASCII file.
2. Passes the ASCII file to a specified program that processes the data and creates a results file for input to EPD.Connect.
3. Reads the results file and initiates an action determined by the data. For a description of the possible actions, see the descriptions of keywords on page 9-9.

Tree Report Command Format

The command line has the following format:

```
program_name report_data_file results_file
```

- Parameter `program_name` typically refers to a script you use to execute other utilities or programs to complete the processing. EPD.Connect checks for a successful return code before displaying the results.
- Parameter `report_data_file` is the ASCII file containing the component data. Refer to “The Report Data File” on page 9-15 for details about the report data file.
- Parameter `results_file` is the file returned to EPD.Connect for display as a text listing or in a tree format. The file can provide a list of components to be highlighted or selected or a list of commands to be returned to CADD5.

Possible return codes are 0 = Success and 1 = Failure. If a results file is present, it can contain error data that is added to the Audit log. The file’s last line is displayed.

Example of Report Generation

The following example produces a weight rollup for an assembly. The weight of an item is specified in a Vault attribute associated with the CADDs part. The command from the menu executes a shell script that uses an `awk` script to perform the summation.

The shell script launched from the menu is as follows:

```
#!/bin/csh
# Example script to produce an assembly
# weight roll-up
if ( $argv != 2 ) then
    echo " usage $argv[0] input_file output_file"
    exit
endif
# Produce weight roll up
cat $argv[1] | awk -F: $CA_REPORT_DIR/weight.awk >
$argv[2]
# end
```

The `awk` script: `weight.awk` is defined as follows:

```
# Initialize variables

BEGIN      filename = "";
           weight = 0;
           total_weight = 0;
           }
{

# Start record located
if ( $1 == "START_ITEM" )
{
    START_ITEM = "YES";
    filename="";
    weight=0;
    next;
}

# End record located print weight of part
if ( $1 == "END_ITEM" && START_ITEM == "YES" )
{
    if ( filename != "" )
    {
        total_weight = total_weight + weight;
    }
}
```

```
                printf ( "%s\t%f\n",filename,weight);
            }
        next;
    }
    if ( $1 == "FILENAME" )
    {
        filename = $2;
    }
    else if ( $1 == "EDMATTRNAME" )
    {
        weight = $4;
    }
}

END {
    printf("Total\t%f\n",total_weight);
}
```

Generating Tree Comparison Reports

Using EPD.Connect you can compare two assembly trees. One of the trees is displayed and the other is stored in memory. The compare operation color codes the text labels of the tree components. Color indicates which components of the displayed tree are the same, new, or changed, as compared to corresponding components of the stored tree. Refer to the *EPD.Connect User Guide* for information about tree comparisons.

In addition to viewing the comparison results, you may want to produce a report of them. This section explains how you customize a comparison report.

Tree Comparison Report Menu

In the Compare Rules dialog box, the Report list enables you to produce reports based on the results of a comparison of two trees. The file name of the Compare Report menu, which is normally held in the menus directory, is defined by the `CA_COMPARE_REPORT` variable. The default for this variable is as follows:

```
CA_COMPARE_REPORT : $CA_MENUS/ca_compr.men
```

The format of the Compare Report menu is as follows:

`display_text` command keyword

- Parameter `display_text` defines the text displayed in the menu. This text must be padded to 40 characters to prevent the command text from being displayed on the menu.
- Parameter `command` is a command to be executed that you can define using the complete path name or specify relative to the reports (`CA_REPORT_DIR`) directory. For a description of the reports directory, see “Generating Tree Reports” on page 9-10.
- Parameter `keyword` defines the action to be performed with the report results. These actions are described in the table on page 9-9.

Possible return codes are 0 = Success and 1 = Failure. If a results file is present, it can contain error data that is added to the Audit log. The file’s last line is displayed.

Setting Colors to Reflect Comparison Results

You can set the color of component text to reflect the result of a comparison. The color allows you to browse the displayed tree for differences. The following table describes these variables.

Table 9-7 Environment Variables for Setting Colors for Comparison Results

Variable	Description	Value
<code>CA_COMPARE_NEW</code>	Component text color for new items	1
<code>CA_COMPARE_CHANGED</code>	Component text color for changed items	9
<code>CA_COMPARE_SAME</code>	Component text color for unchanged items	11

Setting Comparison Tolerances

In performing a comparison, EPD.Connect permits a tolerance on certain attributes. These tolerances affect the comparison of attributes that specify the position and orientation of an item. You use them generally for trees

related to CADDs CAMU databases. The following table describes the variables specifying the tolerance values.

Table 9-8 Environment Variables for Setting Comparison Tolerances

Variable	Description	Value
CA_ORIENT_XYZ_TOL	Specifies the position attribute tolerance	0.001 (default)
CA_ORIENT_ANG_TOL	Specifies the angle attribute tolerance	0.001 (default)

Displaying Reports

Text reports are displayed in the Report Viewer window, which provides for viewing, scrolling, and printing reports.

The Report Data File

The report data file is an ASCII file that defines an EPD.Connect tree. This file defines the structure of the displayed tree to allow analysis by standard utilities, such as `awk`. The data output for a component depends on the level of attribute querying performed when the tree was loaded.

Records defining attributes are output with a multiple definition. The output file starts with a header block, followed by a number of item blocks. The content of the header block and item blocks are described in the following sections.

Format

In the report data file, data is formatted using a keyword followed by a colon and then its value. Most data items are output with one data item per line:

`keyword: data_values`

The keywords used in the report data file are the same as those displayed in EPD.Connect in its Component Status reporting. For more information about Component Status reporting, refer to the *EPD.Connect User Guide*.

Header Block

The file starts with a header block that contains data on the name, revision, type, local directory, and date. The header block begins with a `START_HEAD` record and ends with an `END_HEAD` record, as follows:

```
START_HEAD  
ASSYNAME : TEST . ASSEMBLY  
.  
.  
.  
.  
END_HEAD
```

Starting and Ending a Header Block

The following keywords define the start and end of a header block.

Table 9-9 Keywords for Defining the Start and End of a Header Block

Keyword	Description
<code>START_HEAD</code>	Begins a header block
<code>END_HEAD</code>	Ends a header block

Defining the Assembly

The following keywords introduce values that define the assembly. They are included in the header block.

Table 9-10 Keywords for Defining the Assembly (Header Block)

Keyword	Description
<code>ASSYNAME</code>	Assembly name
<code>ASSYREV</code>	Assembly revision code
<code>ASSYTYPE</code>	Assembly type
<code>ASSYFILE</code>	Assembly local directory
<code>DATE</code>	Assembly time and date

Item Block

A block of data corresponding to an individual component begins with a `START_ITEM` record and ends with an `END_ITEM` record. All records between `START_ITEM` and `END_ITEM` apply to one component. The components for which data is output are determined by a keyword.

Starting and Ending an Item Block

The following keywords define the start and end of an item block.

Table 9-11 Keywords for Defining the Start and End of an Item Block

Keyword	Description
START_ITEM	Begins an item block
END_ITEM	Ends an item block

Defining the Item

The following keywords introduce values that define the item and are included in the item block.

Table 9-12 Keywords for Defining the Item (Item Block)

Keyword	Description
ROW	Item row number
COMPONENT-NAME	Component name
CLASS-NAME	Component class name
INSTANCE	Component instance value
PARENT-NAME	Item's parent component
ORIENTATION	Orientation of referenced part
GLOBAL_ORIENTATION	Global orientation of referenced part
COMP-ORDER	Order of item within sub-assembly

Example of an Item Block

A typical item block is the following:

```

START_ITEM
ROW: 4
COMPONENT-NAME: engine_1
CLASS-NAME: engine
INSTANCE: 1
ITEM-NAME: ENGINE.PART
APPLICATION: CADDS
. . . .
ATTRNAME: planned_date: ATTRVALUE: 29/11/96:
        ATTRCLASS: CLASS
EDMATTRNAME: WEIGHT: ATTRVALUE: 120.5
END_ITEM

```

Defining Characteristics of a Component

The following keywords introduce values that define the characteristics of a component on the tree and are included in the item block.

Table 9-13 Keywords for Defining the Component Characteristics (Item Block)

Keyword	Description
SELECTED	Indicates whether or not the component is selected
HIGHLIGHTED	Indicates whether or not the component is highlighted
VISIBLE	Indicates whether or not the component is visible
ACCESS_COLOUR	Color of the component access indicator
BEEN_EDITED	Indicates whether or not the component has been edited

Defining the Part Associated with a Component

The following keywords provide data about the part associated with a component and its entry in the Vault database. They are also included in the item block.

Table 9-14 Keywords for the Part Associated with a Component (Item Block)

Keyword	Description
TYPE	Associated item type
FILENAME	Associated part file name
ITEM-NAME	Associated part file name
REVISION	Revision of associated part
EDM_REVISION	Revision in Vault of associated part
USERID	User ID of associated part
NODE	Component on which the part resides
CLASS	Class of associated part
OWNER	Owner of associated part
STATUS	Status code of associated part
SYSTEMTYPE	Systemtype attribute value
USERTYPE	Usertype attribute value
PARTNUMBER	Partno attribute value
GTCODE	Gtcode attribute value
DESCRIPTION	Description attribute value

Table 9-14 Keywords for the Part Associated with a Component (Item Block)

Keyword	Description
CREATEUSER	Create user in Vault database
CREATEDATE	Create date in Vault database
CREATETIME	Create time in Vault database
UPDATEUSER	Update user in Vault database
UPDATEDATE	Update date in Vault database
UPDATETIME	Update time in Vault database
SOURCEDATE	Source date in Vault database
SOURCETIME	Source time in Vault database
MODIFY-DATE	Local modify date of part file
MODIFY-TIME	Local modify time of part file
READ-DATE	Local read date of part file
READ-TIME	Local read time of part file
REGISTERED-READ	Registered read marker
READ-REVISION	Read revision

Defining Attributes for a Component

Attributes associated with a component on the tree are defined in an item block, as follows:

```
ATTRNAME : name : ATTRVALUE : value ->
ATTRCLASS : [ CLASS | INSTANCE ] [ : LINK ]
```

- Parameter **name** is the name of the attribute.
- Parameter **value** is the value assigned to the attribute.
- **CLASS** or **INSTANCE** define the attribute's applicability.
- **LINK** specifies that the attribute is applied to the link line.

Defining Vault Attributes for a Component

Vault attributes associated with a component on the tree are defined in an item block, as follows:

`EDMATTRNAME : edmname : ATTRVALUE : edmvalue`

- Parameter `edmname` is the name of the attribute.
- Parameter `edmvalue` is the value assigned to the attribute.

Please note: The assembly and the Vault attributes are output with additional keywords `ATTRVALUE` and `ATTRCLASS` specified on the same line, each separated by a colon from the previous item.

Example of a Report Data File

For an example of a report data file, see the following directory:

`$EPD_HOME/data/reports`

Setting Event Triggers for EPD.Connect

This chapter describes how to set event triggers. The following topics are presented:

- About Event Triggers
- Location of Event Triggers
- Extract Triggers
- Create Triggers
- Editing Event Triggers

About Event Triggers

EPD.Connect uses triggers to force the occurrence of particular actions or checks when data is modified or transferred. These triggers are performed by an external script that returns a code 0 or 1. A 0 code indicates that the event can proceed. Triggers are currently used for the following:

- Commands for adding, changing, and cutting components on a configuration tree
- File, part, product structure, assembly drawing, and assembly transfers to Vault

The trigger programs are defined in the extract and create files. The extract file is activated when a file is transferred to Vault. The create file is activated when a configuration tree is saved.

For product structure files and CADDs assemblies, triggering explodes the data into a Vault attribute file. You can use triggering to automate additional data validation checks and expanded explosions of part and file attributes.

Location of Event Triggers

The triggers applied by EPD.Connect during the transfer of a file to Vault are usually located in its extract directory. You can specify this directory using the `CA_EXTRACT` variable. This directory is normally defined in EPD.Connect's data directory, as follows:

```
CA_EXTRACT : $CA_DATA/extract
```

The variables described in the following table define directories for the file, part, product structure, CAMU assembly, and assembly drawing triggers.

Table 10-1 Environment Variables for Defining Trigger Directories

Variable	Directory Description	Value
<code>CA_EXTRACT_ASSY_DIR</code>	For CAMU assembly trigger programs	<code>\$CA_EXTRACT/camu</code>
<code>CA_EXTRACT_PS_DIR</code>	For product structure trigger programs	<code>\$CA_EXTRACT/ps</code>
<code>CA_EXTRACT_PART_DIR</code>	For part trigger programs	<code>\$CA_EXTRACT/part</code>

Table 10-1 Environment Variables for Defining Trigger Directories

Variable	Directory Description	Value
CA_EXTRACT_FILE_DIR	For file trigger programs	\$CA_EXTRACT/file
CA_EXTRACT_ADRAW_DIR	For assembly drawing trigger programs	\$CA_EXTRACT/adraw

These directories usually reference the extract directory, for example:

```
CA_EXTRACT_PART_DIR : $CA_EXTRACT/PART
```

The other variables are similarly defined, by default, to reside in directories `part`, `ps`, `assy`, and `adraw`.

These variables reside in the template `epdconn.ini` file. They can be set in the `cvepd.ini` or sourced `epdconn.ini` file.

Extract Triggers

An extract trigger is applied when data is stored, updated, or replaced to Vault. The format for the command line is as follows:

```
extract pathname item_name [item_rev] type output_file
```

These directories usually reference the extract directory.

Table 10-2 Extract Trigger Command Parameters

Parameter	Description
pathname	The path to locate the item.
item_name	The name of the part or file in CGOS format.
item_rev	An optional Vault revision code.
type	The type of output file. The possible types are BAG (a Vault attribute) or PS (a PS file for assemblies only).
output_file	The output file.

Create Triggers

A create trigger is relevant only when product structures and CADDs CAMU assemblies are saved. The format for the command line is as follows:
`create pathname config_name [config_rev] type input_file`

Table 10-3 Create Trigger Command Parameters

Parameter	Description
pathname	The path where the configuration is created.
config_name	The name of the configuration to be created.
type	The type of input file. The possible types are BAG (a Vault attribute), REPORT, or PS (a PS file for assemblies only).
input_file	The input data file.

Please note: A return code of 1 prevents the transfer from taking place.

Editing Event Triggers

You can use the `CA_CREATE_TRIGGER` variable to define the program or script that is called when a component in the tree is added, changed, or deleted. The format for the program command line is as follows:
`trigger function attribute_file config_name config_rev ->
component parent [selscope [name[revision]]]`

Table 10-4 Trigger Function Command Parameters

Parameter	Description
function	The name of the calling function. The calling function is ADD, CHANGE, or CUT.
attribute_file	A file containing the component attributes. After the trigger script is executed and control returns to EPD.Connect, the attribute file may contain a list of class or instance attributes that are to be applied to the component. A return code of 1 prevents the edit operation from taking place. The attribute file has the following format: <code>attribute name:CLASS INSTANCE:value</code>
config_name	The name of the active configuration.
config_rev	The revision of the active configuration.
component	The selected component.
parent	The parent of the selected component.

Table 10-4 Trigger Function Command Parameters

Parameter	Description
selscope	The type of associated item (optional).
name	The name of the associated item (optional).
revision	The revision of the associated item (optional).

This chapter presents the following topics:

- System Requirements
- Installing EPD Roles

System Requirements

This section describes system requirements for using EPD Roles.

Please note: Install the operating system as described in the vendor installation documents.

Software Configurations

The following table lists supported operating and window system requirements by platform.

Table 11-1 Operating and Window System Requirements

Platform	Operating System	Window System
Sun	Solaris 2.6	CDE 1.0.1
HP	HP-UX 11.0	Motif 1.2

Hardware Configurations

Minimum configuration requirements for EPD Roles packages depend on the unaccelerated graphics environment.

EPD.Connect UNIX users can operate the 3D Viewer in an unaccelerated graphics environment using X11 mode. XGL mode is not supported in an unaccelerated graphics environment.

Table 11-2 Unaccelerated Graphics Environment for EPD Roles Packages

Platform	Supported Workstation	Memory
Sun	SS5, SS10, SS20, UltraSPARC (all models)	Min/Rec. -128 MB
HP	HP715, HP735,	Min/Rec. -128 MB
HP	C series, J series	Min/Rec. -128 MB

For optimal performance and rendering quality from the 3D Viewer, accelerated graphics and XGL mode are recommended for EPD.Connect UNIX users.

Table 11-3 Accelerated Graphics Environment for EPD Roles Package

Platform	Supported Workstation	Graphics	Memory
Sun	SS5, SS10, SS20, UltraSPARC (all models)	ZX and Creator 3D	Min -128 MB)
HP	HP715, HP735	24Z and 48Z	Min -128 MB
HP	C series, J series, B series	Visualize 24 and 48	Min -128 MB

Disk and Swap Space

The amount of total (combined) disk and swap space recommended for any of the EPD Roles packages is 2 Giga bytes. However, this recommendation is wholly dependent upon the size of databases being managed—including the EPD.Product & Process Management role. Refer to *Installing Optegra Applications* for possible configuration and sizing guidelines.

Sufficient disk space for PDM activities such as using a Distributed Vault must exist somewhere on your network.

License Requirements

FLEXlm is the license management scheme in place for EPD Roles packages. It is used to manage end-user licensing and is administered through normal Customer Services channels.

See *Using the License Manager* to learn about required licenses and their management.

Oracle Runtime License Requirements

One Oracle license is required for every concurrent user of Vault, and/or Distributed Vault.

Installing EPD Roles

If you have purchased the following EPD Roles packages, you must install your applications from both the EPD.Connect and Optegra distribution CD-ROM and the CADD5 distribution CD-ROM using SLIC:

- EPD.Designer
- EPD.Manufacturing Engineer
- EPD.Design Engineer
- EPD.Drafter

Please note: EPD.Product and Process Management is an option under each of these EPD Roles.

If you have purchased the application , EPD.PDM Administrator you must install your applications from the EPD.Connect and Optegra distribution CD-ROM only using SLIC:

Procedure

- 1.** Insert the EPD.Connect and Optegra Applications distribution CD-ROM into the caddy.
- 2.** Start SLIC as described in *Installing Optegra Applications*.
- 3.** Select your specific EPD Role(s) from the onscreen menu.
- 4.** Respond to SLIC prompts as specified in *Installing Optegra Applications*.

If your EPD Role(s) uses CADD5 applications, repeat steps 1 through 4 using the CADD5 distribution CD-ROM and *Installing CADD5 5i*.

- 5.** Install the EPD.Connect and Optegra Applications documentation CD-ROM as described in *Installing Optegra Applications*. Refer to the appropriate user and management documentation for application usage information.

This document provides instructions for configuring EPD.Visualizer. It presents the following topics:

- Setting Required Environment Variables
- Starting EPD.Visualizer
- Setting Options at the Command Line
- Special Workstation Considerations

Please note: Before configuring your EPD.Visualizer application, you must load it. The procedure for loading EPD.Visualizer is documented in *Installing Optegra Applications*.

Setting Required Environment Variables

Set required and optional variables before starting EPD.Visualizer. If you do not, the application sets them for you using defaults. The defaults assume that you are using an accelerated platform.

You can set the variables at the command line, in the `$(EPD_HOME)/scripts/Visualizer` script, or using a file such as `.cshrc`.

Please note: The environment variable statements in this document are written in the UNIX C shell style.

Installation Directory

This variable enables you to specify the directory where you installed EPD.Visualizer.

```
setenv PV_SYSTEM installDirectory/common
```

Local Unaccelerated Graphics

This variable enables you to run EPD.Visualizer on a local workstation using unaccelerated X11 graphics.

```
setenv HOOPS_PICTURE x11/unix:0.0
```

Remote Workstation and Display

This variable enables you to run EPD.Visualizer on a remote workstation and have the graphics appear on another workstation.

```
setenv DISPLAY node:0  
setenv HOOPS_PICTURE x11/node:0.0
```

Please note: Set `node` to the name of the workstation on which EPD.Visualizer should be displayed.

HP Workstation

This variable enables you to run EPD.Visualizer on an accelerated HP 700 Series workstation (24Z, 48Z) or J-200, C-110 (Visual 24 or Visual 48).

```
setenv HOOPS_PICTURE sbx/unix:0.0
```

This variable enables you to run EPD.Visualizer on an HP 700 Series CRX workstation. Use the workstation setup instructions described in “Special Workstation Considerations” on page 12-9.

```
setenv HOOPS_PICTURE sbx/unix:0.0
```

Sun Workstation

This variable enables you to run EPD.Visualizer on a Sun Sparc ZX workstation with accelerated graphics or Ultra Sparc with Creator 3D.

```
setenv HOOPS_PICTURE xgl/unix:0.0
```

Please note: For OpenGL support on an UltraSPARC with creator 3D or Elite 3D, set the following variable:

```
setenv HOOPS_PICTURE opengl/unix:0.0
```

Compaq Tru64 UNIX Workstation

This variable enables you to run EPD.Visualizer on a Digital Alpha workstation with accelerated graphics.

```
setenv HOOPS_PICTURE opengl/unix:0.0
```

Please note: OpenGL 1.1.1 is the minimum requirement for this platform.

IBM Workstation

This variable enables you to run EPD.Visualizer on an IBM RS6000 workstation with accelerated graphics (500D, 1000).

```
setenv HOOPS_PICTURE opengl/unix:0.0
```

Please note: If your IBM accelerator is a 500, define the following variable:

```
setenv HOOPS_PICTURE gl/unix:0.0
```

Please note: OpenGL 1.1.1 is the minimum requirement for this platform.

SGI Workstation

This variable enables you to run EPD.Visualizer on an SGI IRIX workstation with accelerated graphics.

```
setenv HOOPS_PICTURE opengl/unix:0.0
```

Please note: OpenGL 1.1.1 is the minimum requirement for this platform.

Starting EPD.Visualizer

After you set the applicable environment variables described in “Setting Required Environment Variables” on page 12-2, you are ready to start EPD.Visualizer.

Please note: Add the `$EPD_HOME/scripts` path to your execution path, typically your `PATH` variable. This enables you to invoke the application as shown next. Otherwise you must specify the full path to the EPD.Visualizer executable when you invoke EPD.Visualizer. The Visualizer script resides in `$EPD_HOME/scripts`.

To start EPD.Visualizer without command line options, enter the following command:

```
Visualizer
```

To start EPD.Visualizer with command line options, see “Setting Options at the Command Line” on page 12-4.

Setting Options at the Command Line

You can set specific options at the command line. This enables you to:

- List available command line options
- Select a localized language
- Position the palette or its visibility
- Add graphic files, a metafile, or a list of files
- Select the rendering method

- Turn backface polygon culling on or off
- Select image buffering
- Select the graphics window color

The options are described next.

List Command Line Options

This option enables you to display a list of available command line options.

```
Visualizer -help
```

Specify a Language

This option enables you to specify a language. It is available only if you have a version of EPD.Visualizer that has been localized to the specific language.

```
Visualizer -ln languageName
```

For example:

```
Visualizer -ln french
```

Set Palette Position or Hide

This option enables you to set the palette position to the left or right of the graphics window. It also allows you to hide the palette.

```
Visualizer -mj position
```

Acceptable input at the `position` placeholder is `right`, `left` or `hide`. For example:

```
Visualizer -mj left
```

Add Graphic or Meta Files

This option enables you to add graphic files, one metafile, or a combination of graphic files and one metafile.

```
Visualizer FileNameListing
```

Use a space to separate file names. Provide the appropriate filename extension (.gaf, .gbf, .mf).

Please note: You can load only one metafile at a time.

- To add a single binary graphic file, enter the command as follows:
`Visualizer car.gbf`
- To add a combination of text files and binary files, enter the command as follows:
`Visualizer plane.gaf wing.gbf`
- To add a combination of graphic files and one metafile, enter the command as follows:
`Visualizer wheels.gaf car.mf`

Please note: When you add a combination of graphic files and one metafile, the metafile is loaded first.

Add Files from a List

This option enables you to add files from a list.

```
Visualizer -lf Filename
```

For example:

```
Visualizer -lf graphic.files
```

Set Starting Mode

This option enables you to set the starting mode to scene instead of camera:

```
Visualizer -scene
```

Set Rendering Method

This option enables you to set the rendering method to wireframe, hidden line, flat shading, or smooth shading.

```
Visualizer -r renderingMethod
```

Acceptable input at the `renderingMethod` placeholder is `wire`, `hdl`, `flat`, or `smooth`. For example:

```
Visualizer -r hdl
```

Set Buffering Method

This option enables you to set the buffering method to single or double.

```
Visualizer -b bufferingMethod
```

For example:

```
Visualizer -b double
```

Set Backface Culling

This option enables you to turn backface polygon culling on or off.

```
Visualizer -c on/off
```

For example:

```
Visualizer -c on
```

Display System Configuration

This option enables you to display the EPD.Visualizer system configuration.

```
Visualizer -echo
```

Add a License Path

This option enables you to add a license file path for EPD.Visualizer.

```
Visualizer -license licenseFilePath
```

Set Graphics Window Color

This option enables you to set the graphics window color.

```
Visualizer -gc color
```

For example:

```
Visualizer -gc blue
```

The next table shows the available colors.

Table 12-1 Available Graphics Window Colors

black	slate gray	dark green
snow	light slate gray	dark olive green
ghost white	gray	dark sea green
white smoke	light gray	sea green
gainsboro	navy	medium sea green
floral white	cornflower blue	light sea green
old lace	dark slate blue	pale green
linen	slate blue	spring green
antique white	medium slate blue	lawn green
papaya whip	light slate blue	green
blanched almond	medium blue	chartreuse
bisque	royal blue	medium spring green
peach puff	blue	green yellow
navajo white	dodger blue	lime green
moccasin	deep sky blue	yellow green
cornsilk	sky blue	forest green
ivory	light sky blue	olive drab
lemon chiffon	steel blue	dark khaki
seashell	light steel blue	khaki
honeydew	light blue	pale goldenrod
mint cream	powder blue	light goldenrod yellow
azure	pale turquoise	light yellow
alice blue	dark turquoise	yellow
lavender	medium turquoise	gold
midnight blue	turquoise	light goldenrod
lavender blush	cyan	goldenrod
misty rose	light cyan	dark goldenrod
white	cadet blue	rosy brown
dark slate gray	medium aquamarine	indian red
dim gray	aquamarine	saddle brown

Table 12-1 Available Graphics Window Colors (Continued)

sienna	orange	medium violet red
peru	dark orange	violet red
burlywood	coral	magenta
beige	light coral	violet
wheat	tomato	plum
sandy brown	orange red	orchid
tan	red	medium orchid
chocolate	hot pink	dark orchid
firebrick	deep pink	dark violet
brown	pink	blue violet
dark salmon	light pink	purple
salmon	pale violet red	medium purple
light salmon	maroon	thistle

Special Workstation Considerations

Special workstation considerations are categorized by operating environment.

HP Operating Environment

This section presents HP considerations.

Memory Considerations

If you need to load more than 64 megabytes of graphics data into an EPD.Visualizer session on an HP workstation, increase the `maxdsiz` parameter value using the System Administration Manager (SAM).

Please note: The `maxdsiz` parameter represents the maximum data segment size.

Choose Kernel Configuration > Configurable Parameters in SAM to change the parameter value.

If you leave `maxdsiz` at 64mb and load in graphics data that exceeds this limit, you will run out of physical memory. This is true even if you have more than 64mb of memory on your workstation.

Increase the value of this parameter to 268435456, and then rebuild the kernel using SAM.

Graphics Acceleration

If you are running on an HP 700 Series workstation without an accelerator, but still want faster graphics than X11 provides, follow these steps:

1. Install the HP PowerShade software available from Hewlett Packard.
2. Define the HOOPS_PICTURE environment variable as follows:

```
setenv HOOPS_PICTURE sbx/unix:0.0
```
3. Set your HP VUE Color Use to LOW_COLOR with the HP VUE Style Manager.

This is necessary on workstations that have a limited number of graphic planes. You will need to log out and log in again for the change to take effect.

4. If you are using HP VUE, set the following environment variable:

```
setenv SB_X_SHARED_CMAP True
```

Please note: Steps 3 and 4 are not necessary if you log in using the No Windows option (located on the HP startup screen) and run the standard OSF/Motif window manager (by executing `x11start`).

Compaq Tru64 UNIX Operating Environment

This section presents Compaq Tru64 UNIX considerations.

Screen Considerations

To minimize screen updates after closing or moving dialog boxes, add the following to the `.Xdefaults` file in your local directory:

```
Mwm*saveUnder: True
```

Please note: This may reduce window performance on your workstation.

Movie Player

For the EPD.Visualizer movie player to work properly on Digital Alpha workstations having an L2 accelerator, use the 8-bit color map.

X Server

To set up the X server, you must add the `-vclass0 PseudoColor` command line to the `args` statement line in the following file:

```
/usr/var/X11/Xserver.conf
```

After you do this, restart the X server. This setting does not affect the EPD.Visualizer application.

You must be logged in as `root` to perform this task.

Sun Operating Environment

This section presents Sun considerations.

Graphics Library

On Sun Solaris workstations, EPD.Visualizer is dynamically linked with Sun's XGL 3.0 graphics library. Therefore, you must have XGL 3.0 loaded on your system in order to use the EPD.Visualizer XGL driver. The XGL 3.0 library is typically located in the following directory on Solaris systems:

```
/opt/SUNWits/Graphics-sw/xgl-3.0/lib
```

The EPD.Visualizer script expects this path to correctly set your `LD_LIBRARY_PATH`.

Enable Retained Graphics

To enable retained graphics on a Sun Solaris 2.5.1, running on an UltraSPARC, run the following as `root`:

```
ffbconfig -defoverlay true
```

After you do this, restart the X server.

Please note: This setting disables the EPD.Visualizer movie player, `pvs movie`.

See the UNIX man pages on `ffbconfig` for more information about the `ffbconfig` command.

IBM Operating Environment

This section presents IBM considerations.

Retained Graphics

To enable retained graphics on an IBM AIX accelerated platform, the X server must be run using the default layer of 1. The following two methods are used for checking to see if the layer was set to a value other than 1:

- Run the following command to report the Xserver and all its command line arguments:

```
ps agw | grep /usr/bin/X11/X
```

If `-layer number_value` is not listed, the X server is running the default layer of 1.

- Check the following file to see if the variable `EXTENSIONS` has the `-layer number_value` argument in its list:

```
/usr/lpp/X11/defaults/xserverrc
```

When you are finished, restart the X server.

Default Fonts

You can make the following user interface fonts available on any workstation, using the `xset` command. Execute `xset q` to verify your font search path. Refer to the man pages for the `xset` command for additional information.

```
*helvetica-bold-r-*--14*ISO8859-1  
*helvetica-bold-r-*--18*ISO8859-1  
*new century schoolbook-bold-r-*--12*ISO8859-1  
*new century schoolbook-bold-r-*--24*ISO8859-1  
*courier-bold-r-normal--14-*ISO8859-1
```

EPD.Connect Product Structure File

This appendix describes the product structure file. The following topics are presented:

- About the Product Structure File
- Format of the Product Structure File
- Example of a Product Structure File
- Converting Product Structure Trees to CAMU

About the Product Structure File

One input to EPD.Connect is a formatted ASCII file, called the product structure file. A product structure file defines a hierarchy of components. Each component is defined by a name, may have an optional item associated with it, and has an unlimited number of attributes. The name of the product structure file defaults to `_ps`.

Format of the Product Structure File

The product structure file consists of a number of single-line records. Each record has the following format:

`item_number keyword value`

- Parameter `item_number` is an integer used to group together the entries for a specific component.
- Parameters `keyword` and `value` provide information that defines a component in the tree.

The following table describes valid keywords.

Keyword	Description
COMPONENT-NAME	The component name consists of an ASCII string and no spaces. By default, the component name is structured as CLASS-NAME_INSTANCE.
CLASS-NAME	If CLASS-NAME and INSTANCE are present in the file, they override the value of COMPONENT-NAME.
INSTANCE	If CLASS-NAME and INSTANCE are present in the file, they override the value of COMPONENT-NAME.
PARENT-NAME	The COMPONENT-NAME of the parent, NULL for root.
COMPONENT-ORDER	The order in which child components are displayed (optional).
TYPE	Type of associated item - single character (optional).
ITEM-NAME	Name of associated item - ASCII string (optional).
REVISION	Revision of associated item - ASCII string (optional)
ORIENTATION	Six floating point numbers. X,Y,Z,ANGX,ANGY,ANGZ.
GLOBAL-ORIENT	Six floating point numbers. X,Y,Z,ANGX,ANGY,ANGZ.
ATTRIBUTE	Information about an item attribute.

The valid item types for the associated data are as follows:

- A (CADDs assembly)
- T (Product Structure file)
- P (CADDs and CAD/CAM assemblies/parts)
- F (File)

An `ATTRIBUTE` entry can be applied to most components. The format of an `ATTRIBUTE` entry is as follows:

```
ATTRIBUTE attribute_data
```

The parameter `attribute_data` defines a class attribute or instance attribute, as follows:

```
component name : attribute name : CLASS : value [ : LINK ]
```

```
component name : attribute name : INSTANCE : value [ : LINK ]
```

Please note: Notice the use of the colon (:) as a delimiter.

The item numbers in the file must be sequential, starting at 1, and must increase incrementally through the file for each type of entry.

When configurations are created or edited with EPD.Connect, EPD.Connect checks for duplicate instances of components within the tree. It does not check whether multiple instances of subassemblies are correct.

Example of a Product Structure File

The following example shows the product structure for `psdoc`, which consists of two components, `root_1` component and `comp_1` component with associated CADDs part and attributes.

```
1 COMPONENT-NAME psdoc_1
1 CLASS-NAME psdoc
1 INSTANCE 1
1 PARENT-NAME NULL
1 ITEM-NAME PSDOC
1 TYPE T P S Y
1 COMPONENT-ORDER 1
1 ORIENTATION 0.0 0.0 0.0 0.0 0.0 0.0
2 COMPONENT-NAME root_1
2 CLASS-NAME root
```

```
2 INSTANCE 1
2 PARENT-NAME psdoc_1
2 ITEM-NAME ADJ.ARM
2 TYPE P CADDS N
2 COMPONENT-ORDER 2
2 ORIENTATION 0 0 0 0 0 0
1 ATTRIBUTE root_1:MAKE:INSTANCE:ABC::0:1:1
2 ATTRIBUTE root_1:NAME:CLASS:XYZ:LINK:0:1:1
3 COMPONENT-NAME comp_1
3 CLASS-NAME comp
3 INSTANCE 1
3 PARENT-NAME psdoc_1
3 ITEM-NAME ADJ.RIVET
3 TYPE P CADDS N
3 COMPONENT-ORDER 1
3 ORIENTATION 0 0 0 0 0 0
```

You can create a product structure file by using an external program or by using the Edit menu within EPD.Connect.

Converting Product Structure Trees to CAMU

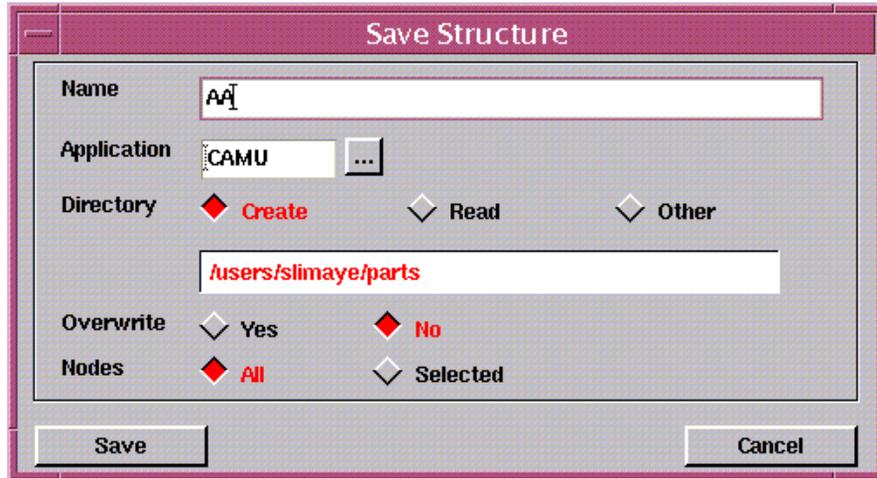
You can convert an EPD.Connect product structure file (`_ps`) to a CAMU assembly database (`_db`) by activating the Product Structure and filing it as a CAMU application type. Follow these steps:

1. Invoke EPD.Connect.
2. Select the Structure tab in the Information Browser.
3. Select File > Open from the top bar.
The Open Configuration property sheet appears.
4. Specify the Product Structure file type in the Application field.

Please note: You can obtain a list of all available `_ps` files by specifying an asterisk in the Name field and selecting the Query option. Selecting an assembly from this list passes the assembly name to the Name field.

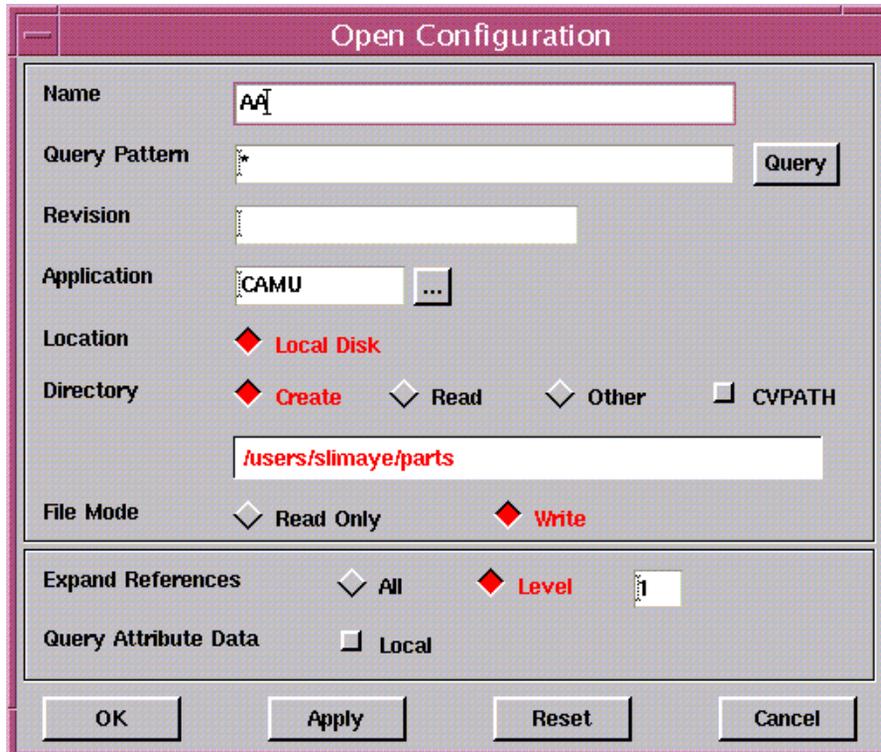
5. Specify the assembly name in the Name field.
6. Specify any other desired settings and select Open to activate the assembly.

7. Select File > Save As. This will launch the Save Structure panel.



8. Specify CAMU in the Application field and click Save to save the file as a CAMU assembly (_db).

9. Open the converted CAMU file in the Product Structure window, using the Open Configuration panel. Click the OK or Apply button in the Open Configuration panel.



EPD.Connect Action Codes

This appendix describes EPD.Connect action codes.

Action Code	Description
6000	Show Component Text window
6040	Tree style fan
6045	Tree style angles
6080	Get
6081	Read
6082	Update
6083	Replace
6084	Sign out
6085	Store
6086	Reset
6087	Copy
6088	Change Attributes
6089	Request Review
6090	Respond to Review
6091	Mark for Archive
6092	Mark for Delete
6093	Unmark
6094	Edit Attributes
6095	Add Read
6096	Delete Read
6097	Refresh Attributes
6098	List
6811	Show the Database Default window
6817	Show the Vault Transfer window

Action Code	Description
6902	Show an overview of the tree display
6915	Show the ADRAW window
6950	Show the Tree Browser window
6960	Show the main tree attribute window
6961	Hide the main tree attribute window
7000	Show the sign-on window
7050	Vault signoff
7100	Show the Open window
7150	Open a new configuration
7190	Show the Save As window
7193	Save the current configuration
7195	Show the Close Reference
7200	Show the Where Used window
7300	Show the Database Transfer window
7400	Show the Memory window
7700	Show the clipboard
7710	Clear the clipboard
7830	Add a component
7832	Copy selected components
7833	Cut selected components
7834	Paste selected components
7835	Undo the previous edit
7836	Redo the previous undo
7837	Swap components
7838	Reorder components
7900	Show the Component Attribute Editor window
8030	Show the Display Parameters window
8100	Show the Status window
8105	Show the Memory Status window
8200	Show the Visible Rules window
8270	Perform a hide
8300	Show the Access Rules window
8370	Perform an access rule evaluation
8380	Clear the access indicator colors

Action Code	Description
8390	Show the color key
8400	Show the Highlight Rules window
8460	Clear the highlighted components
8465	Clear the selected highlighted components
8470	Highlight components
8471	Link selected components
8472	Link highlighted components
8473	Select unhighlighted components
8474	Highlight unselected components
8475	Highlight selected components
8476	Select highlighted components
8479	Perform a Trace Selected
8492	Clear links
8500	Show the Compare Rules window
8550	Perform a Compare
8560	Clear Compare flags
8800	Print an audit log
8899	Change a component
8810	Clear an audit log
8820	Show an audit log
9000	Show a Vault Read message
9110	Show a Vault Send message
9200	Show a Change Password
9960	Clear All
9970	Close the current configuration
9980	Exit the application

EPD.Connect Preconfiguration Checklist

Before you attempt any of the configuration tasks in this book, complete the tasks on this checklist to ensure that the configuration will be successful.

- [] Have system and software backups been performed?
 - __ Operating system
 - __ Optegra software
 - __ Oracle software
 - __ Oracle database(s)

- [] Is the required version of the operating system running?

- [] Are necessary patches installed?

- [] Is Oracle software installed?
 - __ Are `init${ORACLE_SID}.ora` parameters edited?
 - __ Has the directory `${ORACLE_HOME}/dbs` been created?

- [] Are storage pools created (at least two)?
 - __ Are the pools owned by `edm`?
 - __ Are the pools members of group `dba`?
 - __ Is the pools mode `770`?

- [] Has Optegra software been loaded using SLIC?

- [] Have you started Oracle?

Index

Symbols

.caddrsc-local-win 4-20
.caddrsc 4-5, 4-18
.caddrsc-local 4-6
.cshrc 4-17
.Edmgui 4-18
.gaf 4-2
.gbf 4-2
.login 4-17
_db 4-5
_td 4-2

A

ANSPATH 6-2

B

Buttons
 creating a button set 9-4

C

CA_ACCESS_TEMPDIR 8-5
CA_APPLICATION 8-10
CA_APPLICATIONS 8-5
CA_ASSYPREFIX 8-15, 8-16
CA_AUTO_LOAD 6-5
CA_AUTO_SIGNON 6-5
CA_BINDIR 4-4
CA_BUTTON_DIR 8-6, 9-3
CA_BUTTON_FILE 8-6, 9-3
CA_BUTTON_LDIR 8-6, 9-3
CA_BUTTON_LFILE 9-3
CA_CHECK_INSTANCE 8-8
CA_CHECK_TD_SUR 8-12
CA_CLOSE_ICON 8-6
CA_COMPARE_CHANGED 8-2, 9-14
CA_COMPARE_NEW 8-2, 9-14
CA_COMPARE_REPORT 8-5, 9-13
CA_COMPARE_SAME 8-2, 9-14
CA_COMPONENT_DELIM 8-3
CA_COMPONENT_TEXT 8-3
CA_CONFIGURATION 8-10
CA_CREATE_TRIGGER 8-6, 10-4
CA_CREATEMASK 8-6
CA_CUSTOM_MENUS 8-5

CA_CUSTOM_POPUPS 8-5
CA_DATA 4-4, 8-4, 9-4
CA_DESCR_DEFAULT 6-7
CA_DESCR_LABEL 6-7
CA_DESCR_READONLY 6-7
CA_DIR 3-9, 8-4
CA_DISPLAY_PREC 8-11
CA_DYNAMIC_PULLDOWN 8-9
CA_EDIT_ON 8-8
CA_EDM_DEFERRED 8-9
CA_EDM_PASSWORD 6-5
CA_EDM_USER 6-5
CA_ENABLE_LOCKING 8-10
CA_ENABLE_PS 8-7
CA_EXPLODE_CONFIG 8-7
CA_EXTRACT 8-4, 10-2
CA_EXTRACT_ADRAW_DIR 10-3
CA_EXTRACT_ASSY_DIR 10-2
CA_EXTRACT_FILE_DIR 10-3
CA_EXTRACT_PART_DIR 10-2, 10-3
CA_EXTRACT_PS_DIR 10-2
CA_FILE_ENV 8-8
CA_FONT 8-3
CA_FONT_DIR 8-5
CA_GET_OVERWRITE 8-8
CA_GLOBAL_CHANGE_ITEMNAME 8-7
CA_GTCODE_DEFAULT 6-7
CA_GTCODE_LABEL 6-7
CA_GTCODE_READONLY 6-7
CA_HELP 8-5
CA_HIGH_TEMPDIR 8-5
CA_ICONS 8-5, 9-3
CA_LINK_COLOR 8-2
CA_LINK_FOLLOW 8-9
CA_LIST_FONT 8-4
CA_LIST_LOCAL_LONG 8-9
CA_LIST_LOCAL_SHORT 8-9
CA_LOCATION 8-10
CA_LOCK_READ 8-15
CA_LOCKING_DEFAULT 8-10
CA_LT_ACTION 6-9
CA_LT_ACTION_DATE 6-9
CA_LT_ACTIONBSO 6-9
CA_LT_APPLICATION 6-8
CA_LT_ATTRIBUTE 6-9
CA_LT_ATTRSET 6-9
CA_LT_ATTRVALUE 6-9
CA_LT_CLASS 6-8
CA_LT_DB_NAME 6-8
CA_LT_DESCR 6-9
CA_LT_GTCODE 6-9
CA_LT_LOCAL_NAME 6-8
CA_LT_LOCATION 6-8
CA_LT_NODE 6-8
CA_LT_PARTNO 6-9
CA_LT_PROJECT 6-8
CA_LT_REVISION 6-8
CA_LT_STATUS 6-8
CA_LT_STORE 6-9
CA_LT_STORE_DATE 6-9
CA_LT_STOREBSO 6-9
CA_LT_SYSCODE 6-9
CA_LT_SYSTYPE 6-9
CA_LT_UPBSO 6-9
CA_LT_UPDATE 6-9
CA_LT_UPDATE_DATE 6-9
CA_LT_USEDBY 6-8
CA_LT_USERTYPE 6-9
CA_MAX_ITEMS_PER_CACHE 8-11
CA_MENU_FONT 8-3
CA_MENUS 8-5
CA_NODE_BACK_COLOR 8-2
CA_NODE_COLOR 8-2
CA_NODE_SEL_COLOR 8-2
CA_OPEN_DIR_NO 8-10
CA_ORIENT_ANG_TOL 8-12, 9-15
CA_ORIENT_XYZ_TOL 8-12, 9-15
CA_OUTPUT_PREC 8-11
CA_PARTNO_DEFAULT 6-7
CA_PARTNO_LABEL 6-7
CA_PARTNO_READONLY 6-7

- CA_PRESERVE_ATTRS 8-7
- CA_PRINT_COMMAND 8-9
- CA_QUERIES 8-5
- CA_QUERY_DIR 8-5
- CA_QUERY_INFO 8-11
- CA_RB_ENABLED 2-9
- CA_READ_OVERWRITE 8-8
- CA_READPREFIX 8-15, 8-16
- CA_REF_COLOR 8-2
- CA_REPORT_DIR 8-6, 9-10, 9-11, 9-12, 9-14
- CA_REPORT_PREC 8-11
- CA_RESOURCES 8-6
- CA_REV_INTERP 8-8
- CA_REVISION 8-10
- CA_SAVE_TEMPLATE 8-9
- CA_SCRIPTS 4-4, 8-4
- CA_SELECT_MODE 8-11
- CA_SELECT_STATE 8-11
- CA_SMALL_FONT 8-3, 8-4
- CA_STATE_DIR 8-6
- CA_STORE_CLASS 6-6
- CA_STORE_PROJECT 6-6
- CA_STORE_SIGNOUT 6-6
- CA_STRUCTURE_TEXT 8-3
- CA_STYPE_DEFAULT 6-6
- CA_STYPE_LABEL 6-6
- CA_STYPE_READONLY 6-7
- CA_TEMP 3-7, 8-5
- CA_TEMPLATES 8-5
- CA_TRACE_COLOR 8-2
- CA_TREE_FONT 8-4
- CA_TRIGGER 8-6
- CA_UTYPE_DEFAULT 6-6
- CA_UTYPE_LABEL 6-6
- CA_UTYPE_READONLY 6-6
- CA_VIS_TEMPDIR 8-6
- CA_WU_LINK_COLOR 8-2
- CADDS
 - Vault attribute file 10-2
- CADDS CAMU
 - and smaller CADDS windows 4-18
 - configuration 4-5
 - requirement 4-5
- CADDS to STEP and STEP to CADDS translators
 - configuring 5-27
- CADDS to STEP AP203 translator
 - putap203.ini 5-28
- CAMU_EXPAND_REF_ENABLE 8-16
- CATIA
 - configuration 5-19
- cfg1024.ini 3-2, 3-3, 3-9
- cfg1152.ini 3-2, 3-3
- cfg1280.ini 3-2
- cfg800.ini 3-2, 3-3, 3-10, 3-11
- cfgmotif.ini 3-11
- cfgmotif.ini-UNIX 3-2
- cfgtemp.ini 3-10
- cfgwin95 3-3
- cfgwin95.ini 3-11
- cfgwin95.ini—Windows 3-3
- CLASH_CHECK_COMP_READ 8-13
- CLASH_COMP_MAX_CLASHES_FIRST 8-14
- CLASH_FIND_AGAIN_IF_COM_DIS 8-13
- CLASH_IGNORE_DUPLICATE 8-13
- CLASH_IGNORE_FILE_ERRORS 8-13
- CLASH_MEMORIZE_EXTENTS 8-12
- CLASH_OPTIMIZE_ZONING 8-13
- CLASH_ZONING_NOLAY 8-13
- CMOM 2-7
- CMOM_CLIENT_TIMEOUT 5-10
- CMOM_DISPLAY 5-10
- CMOM_DOMAIN 5-10
- Colors
 - reflecting comparison results 9-14
 - setting 8-2
- Comparison
 - reporting results 9-13
 - setting colors to reflect results 9-14
 - setting tolerances 9-14
- Component selection

- controlling 8-11
- Component text
 - setting 8-3
- Configuring EPD.Connect 3-5
- Custom menus
 - command format 9-8
- Customization
 - ptcarb.ini file 5-16
- cvepd.ini 2-2, 2-6, 2-7, 3-1, 3-2, 3-3, 3-5, 3-6, 3-7, 3-8, 3-9, 3-10, 3-11, 4-2, 4-4
- cvobj Component 5-6

D

- DATA_DIRECTORY 2-4
- ddinstall
 - Scripts
 - ddinstall 4-8
- DDL_PATH 5-9
- Directories
 - defining for applications 8-14
 - reports directory 9-11
 - specifying primary directories 8-4
 - specifying subdirectories 8-5
- DISPLAY 5-10
- Documentation, printing from
 - Portable Document Format (PDF) file xxi
- dtconfig 2-10
- dtconfiginfo.txt 2-10

E

- EDM_AVAILABLE 2-9
- edmdvinstall 4-8
- edminstall 4-8
- edmosrv 6-2
- Enabling Nested Reference
 - Assemblies 8-1

- Environement variables
 - EPD_HTMLBROW_DIR 2-7
 - EPD_HTMLBROW_NAME 2-7
 - PVS_PICTURE 2-6
- Environment variable
 - CMOM_CLIENT_TIMEOUT 5-10
 - CMOM_DISPLAY 5-10
 - CMOM_DOMAIN 5-10
 - DATA_DIRECTORY 2-4
 - DDL_PATH 5-9
 - DISPLAY 5-10
 - EPD_HOME 5-10
 - MEDPROJ_PATH 5-9
- Environment
 - variableEDM_AVAILABLE 2-9
- Environment variables
 - ANSPATH 6-2
 - CA_ACCESS_TEMPDIR 8-5
 - CA_APPLICATION 8-10
 - CA_APPLICATIONS 8-5
 - CA_ASSYPREFIX 8-15
 - CA_AUTO_LOAD 6-5
 - CA_AUTO_SIGNON 6-5
 - CA_BINDER 4-4
 - CA_BUTTON_DIR 8-6
 - CA_BUTTON_FILE 8-6
 - CA_BUTTON_LDIR 8-6
 - CA_BUTTON_LFILE 8-6
 - CA_CHECK_INSTANCE 8-8
 - CA_CHECK_TD_SURF 8-12
 - CA_CLOSE_ICON 8-6
 - CA_COMPARE_CHANGED 8-2
 - CA_COMPARE_NEW 8-2
 - CA_COMPARE_REPORT 8-5
 - CA_COMPARE_SAME 8-2
 - CA_COMPONENT_DELIM 8-3
 - CA_COMPONENT_TEXT 8-3
 - CA_CONFIGURATION 8-10
 - CA_CREATE_TRIGGER 8-6
 - CA_CUSTOM_MENUS 8-5
 - CA_CUSTOM_POPUPS 8-5

CA_DATA 4-4, 8-4
CA_DESC_LABEL 6-7
CA_DESC_READONLY 6-7
CA_DIR 3-9, 8-4
CA_DISPLAY_PREC 8-11
CA_DYNAMIC_PULLDOWN 8-9
CA_EDIT_ON 8-8
CA_EDM_DEFERRED 8-9
CA_EDM_PASSWORD 6-5
CA_EDM_USER 6-5
CA_ENABLE_LOCKING 8-10
CA_ENABLE_PS 8-7
CA_EXPLODE_CONFIG 8-7
CA_EXTRACT 8-4
CA_FILE_ENV 8-8
CA_FONT 8-3
CA_FONT_DIR 8-5
CA_GET_OVERWRITE 8-8
CA_GLOBAL_CHANGE_ITEMNAME
8-7
CA_GTCCODE_DEFAULT 6-7
CA_GTCCODE_LABEL 6-7
CA_GTCCODE_READONLY 6-7
CA_HIGH_TEMPDIR 8-5
CA_ICONS 8-5
CA_LINK_COLOR 8-2
CA_LINK_FOLLOW 8-9
CA_LIST_FONT 8-4
CA_LIST_LOCAL_LONG 8-9
CA_LIST_LOCAL_SHORT 8-9
CA_LOCATION 8-10
CA_LOCK_READ 8-15
CA_LOCK_REAL 8-15
CA_LOCKING_DEFAULT 8-10
CA_LT_ACTION 6-9
CA_LT_ACTION_DATE 6-9
CA_LT_ACTIONSO 6-9
CA_LT_APPLICATION 6-8
CA_LT_APROJECT 6-8
CA_LT_ATTRIBUTE 6-9
CA_LT_ATTRSET 6-9
CA_LT_ATTRVALUE 6-9
CA_LT_CLASS 6-8
CA_LT_DB_NAME 6-8
CA_LT_DESC 6-9
CA_LT_GTCCODE 6-9
CA_LT_LOCAL_NAME 6-8
CA_LT_LOCATION 6-8
CA_LT_NODE 6-8
CA_LT_PARTNO 6-9
CA_LT_REVISION 6-8
CA_LT_STATUS 6-8
CA_LT_STORE 6-9
CA_LT_STORE_DATE 6-9
CA_LT_STOREBSO 6-9
CA_LT_SYSCODE 6-9
CA_LT_SYSTYPE 6-9
CA_LT_UPBSO 6-9
CA_LT_UPDATE 6-9
CA_LT_UPDATE_DATE 6-9
CA_LT_USEDBY 6-8
CA_LT_USERTYPE 6-9
CA_MENU_FONT 8-3
CA_MENUS 8-5
CA_NODE_BACK_COLOR 8-2
CA_NODE_COLOR 8-2
CA_NODE_SEL_COLOR 8-2
CA_OPEN_DIR_NO 8-10
CA_ORIENT_ANG_TOL 8-12
CA_ORIENT_XYZ_TOL 8-12
CA_OUTPUT_PREC 8-11
CA_PARTNO_DEFAULT 6-7
CA_PARTNO_LABEL 6-7
CA_PARTNO_READONLY 6-7
CA_PRESERVE_ATTRS 8-7
CA_PRINT_COMMAND 8-9
CA_QUERIES 8-5
CA_QUERY_DIR 8-5
CA_QUERY_INFO 8-11
CA_RB_ENABLED 2-9
CA_READ_OVERWRITE 8-8
CA_READPREFIX 8-15
CA_REF_COLOR 8-2, 8-5

- CA_REPORT_DIR 8-6
- CA_REPORT_PREC 8-11
- CA_RESOURCES 8-6
- CA_REV_INTERP 8-8
- CA_REVISION 8-10
- CA_SAVE_TEMPLATE 8-9
- CA_SCRIPTS 4-4, 8-4
- CA_SELECT_MODE 8-11
- CA_SELECT_STATE 8-11
- CA_SMALL_FONT 8-4
- CA_STATE_DIR 8-6
- CA_STORE_CLASS 6-6
- CA_STORE_PROJECT 6-6
- CA_STORE_SIGNOUT 6-6
- CA_STRUCTURE_TEXT 8-3
- CA_STYPE_DEFAULT 6-6
- CA_STYPE_LABEL 6-6
- CA_TEMP 3-7, 8-5
- CA_TEMPLATES 8-5
- CA_TRACE_COLOR 8-2
- CA_TREE_FONT 8-4
- CA_TRIGGER 8-6
- CA_UTYPE_DEFAULT 6-6
- CA_UTYPE_LABEL 6-6
- CA_UTYPE_READONLY 6-6
- CA_VIS_TEMPDIR 8-6
- CA_WU_LINK_COLOR 8-2
- CVCREATESTEP 5-25
- EDM_AVAILABLE 2-9
- EPD_HOME 2-2
- EPD_HTMLBROW_CMDFMT 2-7
 - for configuring EPD.Connect 3-5
 - servercharacterse 2-10
 - setting 3-5, 3-6
- EPD Roles
 - disk and swap space 11-3
 - hardware configurations 11-2
 - license requirements 11-3
 - loading 11-4
 - software configurations 11-2
 - system requirements 11-2

- EPD.Visualizer
 - configuration 4-3
- EPD_HOME 2-2, 5-10
- EPD_HTMLBROW_CMDFMT 2-7
- EPD_HTMLBROW_DIR 2-7
- EPD_HTMLBROW_NAME 2-7
- epdconn.in 3-11
- epdconn.ini 3-2, 3-3, 3-6, 3-7, 3-9, 3-10, 3-11, 8-2, 10-3
- Event triggers
 - editing 10-4
 - location of 10-2
 - setting 10-2
- explorer.in 2-2, 3-6, 3-7
- explorer.ini 3-2, 3-3, 3-9, 3-11, 4-2, 4-4
- Extracting Orientation Information 5-16

F

- File banes
 - cvepd.ini 3-2
- File names
 - .caddrsc-local-win 4-20
 - .caddsrc 4-18
 - .caddsrc-local 4-6
 - .cshrc 4-17
 - .Edmgui 4-18
 - .gaf 4-2
 - .gbf 4-2
 - .login 4-17
 - _db 4-5
 - _td 4-2
 - caddsrc-local 4-5
 - cfg1024.ini 3-2, 3-3, 3-9
 - cfg1152.ini 3-2, 3-3, 3-10
 - cfg1280.ini 3-2, 3-3, 3-9, 3-11
 - cfg800.in 3-10
 - cfg800.ini 3-2, 3-3, 3-11
 - cfgmotif.in 3-11

cfgmotif.ini—UNIX 3-2
 cfgwin95.ini 3-11
 cvepd.in 2-9
 cvepd.ini 2-2, 2-6, 2-7, 3-1, 3-3, 3-5, 3-6,
 3-7, 3-8, 3-9, 3-10, 4-2, 4-4
 dtconfig 2-10
 dtconfiginfo.txt 2-10
 epdconn.ini 3-2, 3-3, 3-6, 3-7, 3-8, 3-9,
 3-11
 explorer.in 3-7
 explorer.ini 2-2, 3-2, 3-3, 3-6, 3-9, 3-11,
 4-2, 4-4
 Get.pl 5-11
 getap203.ini 5-27
 icfg1280.ini 3-2
 infobrowser.ini 3-3, 3-4
 medusaif 5-6
 medusarb.ini 5-6
 nsm.conf 4-17
 pm.config 2-8, 4-17, 5-22, 5-29
 proeif 5-12
 ptcarb.ini 5-16
 Read.pl 5-11
 registry.ini 3-3, 3-4
 Replace.pl 5-11
 Signoff.pl 5-11
 Signon.pl 5-11
 Store.pl 5-11
 template.caddsrc 2-4
 Update.pl 5-11
 vault.ini 3-3, 3-4
 Functional Components 5-6

G

Geometric accuracy
 controlling positioning 8-11
 controlling storage precision 8-11
 Get.pl 5-11

I

infobrowser.ini 3-3, 3-4
 Initialization files 3-6, 3-7
 cfg1024.ini 3-2, 3-3, 3-9
 cfg1280.ini 3-2, 3-3, 3-9, 3-10
 cfg800.ini 3-2, 3-3, 3-10
 cfgmotif.ini 3-2, 3-10
 cfgtemp.ini 3-10
 cfgwin95.ini 3-2, 3-3, 3-10
 cvepd.ini 2-6, 3-5, 3-6, 3-8
 epdconn.ini 3-5, 3-6, 3-9, 8-2, 10-3
 explorer.ini 3-9, 4-4
 Installing EPD Roles 11-2
 Installing EPD.Connect
 on the same system as Vault 4-8

L

Local defaults
 modifying 3-6
 Locale for a Language 2-11

M

Mask 8-6
 MEDPROJ_PATH 5-9
 Medusa
 executable 5-6
 initialization file 5-6
 medusaif 5-6
 medusarb.ini 5-6
 Menus
 Compare Report menu 9-14
 Migration of Data for Optegra
 Interface for
 Pro/ENGINEER 5-15

N

Navigator
 upgrading from 7-2
navinstall 4-7, 4-8, 4-9
navsetup 4-8, 4-17
Nested Reference Assembly 8-16
 enabling 8-16
nsm.config 6-3

O

Open Configuration window
 setting up 8-10

P

Part_type 5-15
 PTCASM 5-15
 PTCDRW 5-15
 PTCFRM 5-15
 PTCLAY 5-15
 PTCPRT 5-15
pm.config 4-17, 5-22, 5-29, 6-3
 environment variable 6-2
pm.config file
 identifying Vault access 6-3
Printing documentation from
 Portable Document Format
 (PDF) file xxi
Product Background 5-6
Product Structure file (_ps)
 example A-3
 format A-2
 purpose A-2
proeif 5-12
ptcarb.ini 5-16
PTCRB_REPOSIT 5-16
 .optnav file 5-16

Navigator startup script 5-16
PVS_PICTURE 2-6

R

Read.pl 5-11
Refresh tool
 exiting 7-2
 restarting 7-2
 running 7-2
registry.ini 3-3
Replace.pl 5-11
Report data file
 content 9-15
 format 9-15
 purpose 9-15
Reports
 displaying 9-15
 tree comparison 9-13
 tree report 9-10
Revised label
 displaying 6-10

S

Scripts
 edmdvinstall 4-8
 edminstall 4-8
 navinstall 4-7, 4-8, 4-9
 navsetup 4-8, 4-17
servercharacterstet 2-10
setenv command
 defining environment variables 3-5,
 3-6
Signoff.pl 5-11
Signon.pl 5-11
Simultaneous EPD.Connect
 sessions 3-7
STEP AP203 to CADD5

getap203.ini 5-28
Store.pl 5-11

T

Tessellation file
 controlling the transfer of 8-12
Text fonts
 setting for interface 8-3
Timeouts, controlling 2-7
Tree comparison report
 generating 9-13
Tree report
 generating 9-10

U

Update.pl 5-11

V

Vault
 automatic startup 6-5
 connecting to 2-9
 launching 6-5
 PC installation 2-9
 pm.config 4-17, 6-2
 setting transfer triggers 10-2
 Win95 2-10
Vault Query Options window
 setting defaults 6-8
Vault Store window
 setting defaults 6-6
Vault Transfers window
 setting defaults 6-8
vault.ini 3-3, 3-4

