

EMC Solutions Enabler
Version 6.0

INSTALLATION GUIDE

**P/N 300-002-289
REV A01**

EMC Corporation
Corporate Headquarters:
Hopkinton, MA 01748-9103
1-508-435-1000
www.emc.com

Copyright © 2002, 2003, 2004, 2005 EMC Corporation. All Rights Reserved.

Printed January, 2005

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

Trademark Information

EMC², EMC, Symmetrix, Celerra, CLARiiON, CLARAlert, Documentum, HighRoad, Legato, Navisphere, PowerPath, ResourcePak, SnapView/IP, SRDF, TimeFinder, VisualSAN, and where information lives are registered trademarks and EMC Automated Networked Storage, EMC ControlCenter, EMC Developers Program, EMC OnCourse, EMC Proven, EMC Snap, Access Logix, AutoAdvice, Automated Resource Manager, AutoSwap, AVALONidm, C-Clip, Celerra Replicator, Centera, CentraStar, CLARevent, Connectrix, CopyCross, CopyPoint, DatabaseXtender, Direct Matrix, Direct Matrix Architecture, EDM, E-Lab, Enginuity, FarPoint, FLARE, GeoSpan, InfoMover, MirrorView, NetWin, OnAlert, OpenScale, Powerlink, PowerVolume, RepliCare, SafeLine, SAN Architect, SAN Copy, SAN Manager, SDMS, SnapSure, SnapView, StorageScope, SupportMate, SymmAPI, SymmEnabler, Symmetrix DMX, Universal Data Tone, and VisualSRM are trademarks of EMC Corporation. All other trademarks used herein are the property of their respective owners.

Preface	ix
----------------------	----

Chapter 1 Pre-Install Considerations

Introduction	1-2
Before You Begin	1-3
General Tasks	1-3
Windows Specific Tasks	1-3
Client or Server Installation.....	1-4
Remote Connection	1-4
Client/Server Security	1-5
Client/Server System Installation	1-5

Chapter 2 Installation

Installing Solutions Enabler on UNIX	2-2
Step 1: Mount the CD-ROM	2-2
Step 2: Run the Install Script	2-3
Step 3: Choose the Installation Directories	2-3
Step 4: Select the Product to Install	2-5
Step 5: Select Options	2-6
Step 6: Complete the Installation.....	2-10
Installing Solutions Enabler on Windows	2-12
Installing Solutions Enabler on OpenVMS	2-20

Chapter 3 Installation Follow-Up

Enabling Your Software	3-2
License Keys	3-2
Enabling Components.....	3-3

Initial Steps for Post Solutions Enabler Install	3-5
Build the SYMAPI Database	3-5
Set Environment Variables.....	3-5
Set Access Permissions to Directories	3-5
Starting the SCSI Generic Driver.....	3-5
Setting the CLI Path.....	3-6
Setting the Online Help Path	3-7
Symmetrix Gatekeeper Devices.....	3-8
Choosing a Gatekeeper.....	3-8
Using the gkavoid and gkselect Files	3-9
Using a Dedicated Gatekeeper	3-10
Sizing a Gatekeeper.....	3-10
Database and Gatekeeper Locking.....	3-11
Semaphore Requirements on UNIX	3-11
Meeting Semaphore Requirements.....	3-12
Refreshing the Semaphores.....	3-12
De-Allocating Semaphores.....	3-12
Semaphore Identifier.....	3-12
OpenVMS Locking	3-13
Windows Locking.....	3-13
Avoidance and Selection Files.....	3-14
Editing and File Format.....	3-14
gkavoid and gkselect	3-14
inqfile.....	3-15
symavoid	3-15
Options File	3-16
Editing the Options File.....	3-16
Removing Default Options	3-18
SYMCLI Through a Remote Server.....	3-19
Securing Remote Access	3-19
Editing the netcnfg File.....	3-23
Configuring the Trusted-User Host Access File	3-24
Registering the Server in the Windows Service Control Manager	3-24
Starting the Server	3-25
Setting Environment Variables for Remote Access	3-26
Stopping the Server.....	3-27
Uninstalling the Server (Windows)	3-27
Oracle Multiple Instances through Remote Server.....	3-28
Client/Server RDBMS Environment Variable Behavior ..	3-29
Setting Up Daemons for Distributed Application Support.....	3-30
Starting Daemons	3-31
Stopping Daemons	3-31

Viewing Daemons	3-32
Configuring Daemons to Auto-Start on Boot	3-32
Building a daemon_users File	3-32
Controlling Daemon Behavior	3-34
Base Daemon Support	3-35
Configuring and Managing the SMI Provider	3-39
Starting and Stopping the Pegasus CIM Server	3-39
Controlling the EMC SMI Provider at Runtime	3-40
Configuring and Managing CLARiiON Arrays	3-41
Optional SSL Configuration	3-42
Optional SLP Configuration	3-45

Chapter 4 Uninstalling Solutions Enabler

Overview	4-2
Stopping the Application Processes	4-2
Uninstalling the Software	4-3
Uninstalling Solutions Enabler from UNIX	4-4
Uninstalling Solutions Enabler from Windows	4-6
Uninstalling Solutions Enabler from OpenVMS	4-7
Rolling Back an Upgrade	4-8

Appendix A Host Specific Issues

HP-UX Issues	A-2
Semaphores for Database and Gatekeeper	A-2
Creating Pseudo-devices for Gatekeepers and BCVs	A-3
Sun Solaris Issues	A-6
Compaq/Digital UNIX Issues	A-7
Compaq TruCluster Server	A-7
Digital OpenVMS Issues	A-9
IBM AIX Issues	A-13
Memory Access	A-13
Oracle Database Mapping	A-13
BCV Devices Lost after Reboot	A-14
Linux Issues	A-15
Mounting a CD Over a Network	A-15
Installing the Kernel Patch	A-15
Windows Issues	A-16

Appendix B Creating Daemon Auto-Start Scripts

- Creating an Auto-Start Script for Solaris B-2
- Creating an Auto-Start Script for Linux B-3
- Creating an Auto-Start Script for HP-UX B-4
- Creating an Auto-Start Script for IBM AIX B-6
 - Run Level Scripting — Installed B-6
 - Run Level Scripting — Not Installed..... B-7
- Creating an Auto-Start Script for Windows B-8

Appendix C Solutions Enabler Directories

- UNIX Directories C-2
- Windows Directories C-3

Index..... i-1

Tables

2-1	Installation Temporary Files	2-10
3-1	Solutions Enabler License Keys	3-2
3-2	Optional Behavior Parameters for Options File	3-16
3-3	Supported Solutions Enabler Daemons	3-30
3-4	Base Daemon Optional Behavior Parameters	3-38
3-5	EMC SMI Provider Properties	3-40
3-6	CIM Configuration File	3-43
A-1	OpenVMS Return Code Interpolation Table	A-9
C-1	UNIX Directories	C-2
C-2	Windows Directories	C-3

As part of its effort to continuously improve and enhance the performance and capabilities of the EMC product line, EMC periodically releases new versions of both the EMC Enginuity Operating Environment and EMC Solutions Enabler. Therefore, some functions described in this guide may not be supported by all versions of Enginuity or Solutions Enabler currently in use. For the most up-to-date information on product features, see your product release notes.

If a Solutions Enabler feature does not function properly or does not function as described in this guide, please contact the EMC Customer Support Center for assistance.

This guide provides installation procedures for installing the EMC Solutions Enabler software for your specific platform. The EMC Solutions Enabler software provides your host system with an API shared library and a special command set that comprises the Symmetrix Command Line Interface (SYMCLI). (For z/OS types, only the SYMAPI server is available.)

Organization

Here is an overview of where information is located in this guide.

Chapter 1, *Pre-Install Considerations*, explains the tasks that you should perform before installing Solutions Enabler.

Chapter 2, *Installation*, explains how to install Solutions Enabler.

Chapter 3, *Installation Follow-Up*, provides the follow-up procedures for a Solutions Enabler installation.

Chapter 4, *Uninstalling Solutions Enabler*, explains how to uninstall Solutions Enabler.

Appendix A, *Host Specific Issues*, lists the host specific requirements for running Solutions Enabler.

Appendix B, *Creating Daemon Auto-Start Scripts*, explains how to set daemons to automatically start when a system is rebooted.

Appendix C, *Solutions Enabler Directories*, contains the directory list for UNIX and Windows installations.

Related Documentation

Related documents include:

- ◆ *EMC Solutions Enabler Version 6.0 Support Matrix*
- ◆ *EMC Solutions Enabler for z/OS Implementation Guide*
- ◆ *EMC TimeFinder/Integration Modules Product Guide*
- ◆ *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide*
- ◆ *EMC Host Connectivity Guides*

Conventions Used in this Manual

The following conventions are used in this manual:

Every use of the word SYMCLI means Solutions Enabler.

Every use of z/OS in text means z/OS and OS/390.

A note calls attention to any item of information that may be of special importance to the reader.



CAUTION

A caution contains information essential to avoid damage or degraded integrity to storage of your data. The caution might also apply to protection of your software or hardware.

Typographical Conventions

This manual uses the following type style conventions in this guide:

bold text	Boldface text provides extra emphasis and emphasizes warnings, and specifies window names and menu items in text.
<i>italic text</i>	<i>Italic text and characters emphasizes new terms, identifies variables in a software syntax (non-literal notation), identifies unique word usage, and applies emphasis in examples and in references to book titles and sections.</i>
fixed space courier font	A fixed space font identifies files and path names, and is used in command line entries, displayed text, or program listings.

Where to Get Help

EMC Solutions Enabler software products are supported directly by the EMC Customer Support Center and the EMC Developers Program.

For Solutions Enabler products, you can contact technical support by calling the EMC Customer Support Center at one of the following numbers:

- United States: (800) 782-4362 (SVC-4EMC)
- Canada: (800) 543-4782 (543-4SVC)
- Worldwide: (508) 497-7901

Follow the voice menu prompts to open a service call and select the applicable product support.

If you are a partner in the EMC Developers Program, refer to the EMC Powerlink Website at:

<http://powerlink.emc.com>

Sales and Customer
Service Contacts

For the list of EMC sales locations, please access the EMC home page at:

<http://emc.com/contact/>

For additional information on the EMC products and services available to customers and partners, refer to the EMC Powerlink Website at:

<http://powerlink.emc.com>

Your Comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Please send a message to **techpub_comments@emc.com** with your opinions of this guide.

Your technical enhancement suggestions for future development consideration are welcome. To send a suggestion, log on to <http://powerlink.emc.com>, follow the path **Support, Contact Support**, and choose **Software Product Enhancement Request** from the Subject menu.

Pre-Install Considerations

This chapter explains the tasks that you should perform before installing Solutions Enabler.

- ◆ Introduction 1-2
- ◆ Before You Begin 1-3
- ◆ Client or Server Installation..... 1-4

Introduction

A Solutions Enabler install provides your host with SYMAPI, CLARAPI, and STORAPI shared libraries for use by Solutions Enabler applications, and the Symmetrix® Command Line Interface (SYMCLI) for use by Storage Administrators and Systems Engineers.

SYMCLI is a specialized library of UNIX formatted commands that can be invoked one at a time. It supports single command line entries and scripts to map and perform control operations on devices and data objects toward the management of your storage complex. It also monitors device configuration and status of devices that make up the storage environment. The target storage environments are typically Symmetrix, but can be CLARiiON® when you have a license and work with the mapping SRM component.

Solutions Enabler includes a monitoring option for situations on certain hosts where you need to limit the SYMCLI actions to monitor mode only and block all SYMCLI activity from executing any control actions or ones affecting any changes to the Symmetrix array.

Before You Begin

Before you begin to install Solutions Enabler, be sure to complete the tasks listed in this section.

General Tasks

The following tasks apply to all supported platforms:

- ☐ Review the requirements listed in the *EMC Solutions Enabler Release Notes*.
- ☐ If you are upgrading from a previous version, verify that all application processes that use the Solutions Enabler libraries and binaries are stopped.

For instructions, refer to *Stopping the Application Processes* on page 4-2.

- ☐ In the event that you would ever want to roll back to a previous version of Solutions Enabler, create copies of the host database and config directories. The location of these directories differ for each platform:

Platform	Default Directories
UNIX	• /var/syampi/db • /var/syampi/config
Windows	• C:\Program Files\EMC\symapi\db • C:\Program Files\EMC\symapi\config
OpenVMS	• SYMAPI\$DB: • SYMAPI\$CONFIG

Windows Specific Tasks

The following tasks are specific to Windows:

- ☐ During the installation process, the installation program will prompt you to install the SYMAPI server. Prepare for this question by reading the information in *Client or Server Installation* on page 1-4.
- ☐ During the installation process, the installation program will prompt you to install the storapi daemon. Prepare for this question by reading the information in *Setting Up Daemons for Distributed Application Support* on page 3-30.

Client or Server Installation

If your computer is locally connected to a Symmetrix array, go to Chapter 2. If your computer is a client or the SYMAPI server, read the following sections.

Remote Connection

You can run SYMCLI as a client to a remote SYMAPI server to manage a remotely-controlled Symmetrix array. Figure 1-1 shows a Symmetrix array in the client/server system.

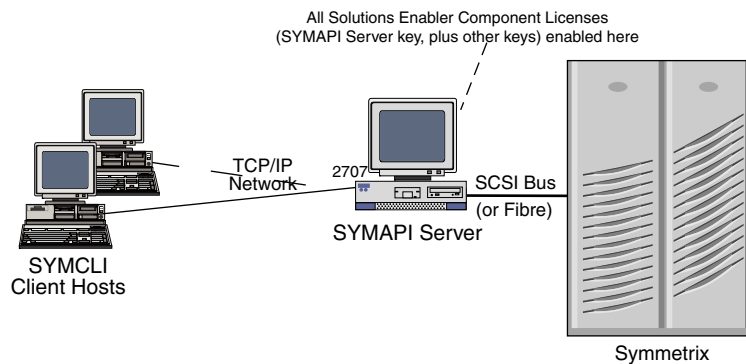


Figure 1-1 Client/Server Connection to Symmetrix

The Solutions Enabler installation program installs the server program (`symapisrv`) in the bin directory. The program also installs the following files in the configuration directory (`/var/symapi/config`):

- ◆ `netcnfg` lists, on both the client and server hosts, the network services available from that local host or from one or more remote hosts.
- ◆ `symapinlck` and `symapislck` are internal files used by the server as a lock file to guarantee a single port listener for a network service. Do not delete or alter these files.
- ◆ `nethost` lists the nodes/users allowed to connect to the server to execute remote SYMAPI functions.
- ◆ `symapisrv_trust.pem` and `symapisrv_cert.pem` are certificate files used by the client to verify the identity of the server for remote sessions.

Client/Server Security

Solutions Enabler uses Secure Socket Layer (SSL) protocol to enable secure communication in a client/server system. Using open source SSL (OpenSSL) technology, the client and server communicate over an authenticated, encrypted connection.

When a client attempts to connect with a server, the two machines exchange a handshake in which they both identify their security expectations and capabilities. If their security capabilities are the same, the two will negotiate the appropriate type of session (secure or non-secure). If their security capabilities are different, either the client or the server will reject the session.

The SYMAPI client and server are initially configured to negotiate both secure and non-secure sessions. This is to facilitate communication between the client and server where the SSL capabilities do not agree. For instructions on modifying this default behavior, refer to *Securing Remote Access* on page 3-19.

Client/Server System Installation

The following information outlines procedures for installing Solutions Enabler in a client/server system:

1. Install Solutions Enabler software in the machine designated as the client, according to the procedures in Chapter 2.
2. Install the same Solutions Enabler software in the machine designated as the server, according to the procedures in Chapter 2. You need to invoke `sym1mf` and apply the SYMAPI Server license key. On Windows servers, you also need to register a SYMAPI service option if you did not use the default installation. For instructions, refer to *Registering the Server in the Windows Service Control Manager* on page 3-24.
3. Edit the `netcnfg` file in the client machine to include the TCP/IP address of the network service. For instructions, refer to *SYMCLI Through a Remote Server* on page 3-19.
4. Issue a `symapisrv start` on the server machine. For instructions, refer to *SYMCLI Through a Remote Server* on page 3-19.
5. Set environment variables `SYMCLI_CONNECT` and `SYMCLI_CONNECT_TYPE`. For instructions, refer to *SYMCLI Through a Remote Server* on page 3-19.

This chapter explains how to install Solutions Enabler.

The procedures in this chapter are organized by operating system.

- ◆ Installing Solutions Enabler on UNIX.....2-2
- ◆ Installing Solutions Enabler on Windows2-12
- ◆ Installing Solutions Enabler on OpenVMS.....2-20

Installing Solutions Enabler on UNIX

This section describes how to install Solutions Enabler in a UNIX environment. The following procedure can be used for either a new installation, or to upgrade an existing installation.

Before starting this procedure, be sure to review Chapter 1 of this document and the *EMC Solutions Enabler Release Notes*.

Step 1: Mount the CD-ROM

To mount the CD-ROM:

If you are installing Solutions Enabler in a Linux on S390 environment, you must mount the CD over a network. For instructions, refer to *Mounting a CD Over a Network* on page A-15.

1. Log onto the host system as `root`.
2. Mount the CD-ROM to a subdirectory (for example, `/cdrom`) by entering the following according to your operating system:

For	Enter
AIX ^a	<code>mount -r -v cdrfs /dev/cd0 /cdrom</code>
Compaq Tru64 UNIX	<code>mount -t cdfs -o [noversion] rrip /dev/disk/cdromxc /cdrom</code>
HP-UX Versions 11.0 and above	<code>mount -F cdfs /dev/dsk/cxtxdx /cdrom</code>
Linux	<code>mount -o ISO9660,ro /dev/cdrom /cdrom</code> Solutions Enabler should not be loaded on the Celerra® File Server Control Station. The Celerra File Server is not a Linux Client.
Solaris	If automounter is running, the CD-ROM mounts unattended. To mount the CD-ROM manually, enter: <code>mount -F hsfs -o ro /dev/dsk/cxtxdxs0 /cdrom</code>
UNIX_SV	<code>mount -F cdfs -o ro,suid,susp,rrip /dev/dsk/c110t2d0s0 /mnt</code>

a. With AIX, you may get a warning if the device and the directory do not have the same permissions. You can usually ignore these warnings.

Step 2: Run the Install Script

Change directory to the location of the Solutions Enabler kit and run the install script by entering the following:

```
cd /<CD-ROM mount point>/UNIX
./emc_install.csh
```

Once the installation program starts, you cannot move it to background.

The script displays the following:

```
#-----
# EMC Installation Manager
#-----
Copyright 2004, EMC Corporation
All rights reserved.

The terms of your use of this software are governed by
the applicable contract.
```

Step 3: Choose the Installation Directories

The default responses to the prompts in this section are shown in brackets [].

1. At the following prompt, enter the location of the Solutions Enabler kit. The kit is usually located in the `UNIX` directory, below the mount point of the CD-ROM (the parent of the `INSTALL` directory).

```
Kit Location [/<kit location>/osl_cd]:
```

It is recommend that you use the default location.

2. At the following prompt, enter an installation directory for the software:

```
Install root directory [/opt/emc]:
```

It is recommended that you install EMC Solutions Enabler on your host's internal disks and not on a Symmetrix device.

3. At the following prompt, enter a working directory where the data and log files will be written:

Working root directory [/usr/emc]:

If the root directory and working directory do not exist, the installation script will automatically create them, as shown below:

```
The install root directory /opt/emc does not exist.  
This directory is necessary for the installation to complete  
successfully.
```

```
It will be created now. Is that OK? [y] y
```

```
./emc_install : Creating Directory      ---- /opt/emc
```

```
The working root directory /usr/emc does not exist.  
This directory is necessary for the installation to complete  
successfully.
```

```
It will be created now. Is that OK? [y] y
```

```
./emc_install : Creating Directory      ---- /usr/emc
```

Step 4: Select the Product to Install

After selecting the installation directories, two lists of Solutions Enabler products display:

- ◆ The first list displays the Solutions Enabler products that the install script detects have already been installed on the host:

```
#-----
# The following HAS BEEN INSTALLED in /opt/emc via the emc_install utility.
#-----
# ITEM   PRODUCT                                     VERSION
#-----
```

- ◆ The second list displays the products available to be installed:

```
#-----
# The following is AVAILABLE TO BE INSTALLED from the selected Kit Location:
#-----
# ITEM   PRODUCT                                     VERSION
#-----
01      EMC Solutions Enabler                        V6.0.0

What would you like to do? Install (i) Uninstall (u) Register (r) Exit (x):
```

The Register option is not available at this time.

To exit the installation, select exit from the install menu by typing: **x**

1. At the following prompt, enter **i** for Install:

```
What would you like to do? Install (i) Uninstall (u) Register (r) Exit (x): i
```

2. At the following prompt, enter the item number **01**:

```
Enter item number (from available) to install: 01
```

Step 5: Select Options

The default responses to the prompts in this section are shown in brackets [].

1. At the following prompt, specify whether to install the Solutions Enabler libraries:

```
Install All Solutions Enabler Shared Libraries and Run
Time Environment? [Y]
```

- A [Y]es response installs *all* the libraries.
- A [N]o response produces the following series of prompts, which allow you to select the libraries to install. If you only want to install Solutions Enabler's core functionality, specify **N[o]** for each of the prompts. Solutions Enabler's core functionality includes symapi [mt], Symmlvm[mt], Storapi [mt], Symapisrv, Storapid, Storcore, stordaemon, and storpds.
 - BaseStor Library Component ? [Y]
A [Y]es response installs StorSil and Storbase. This option provides base storage and host specific functionality, and an interface to storage arrays for features like I/O scan, device listings, statistics, and showings.
 - StorCtrl Library Component ? [Y]
A [Y]es response installs StorSil, Storbase, and Storctrl. This option provides the same functionality as the BaseStor option (explained earlier), but includes storage control functionality for features like Snap, device masking, and device mirroring.
 - BaseMapping Library Component ? [Y]
A [Y]es response installs Stormap, which provides Storage Resource Management (SRM) functionality.

2. Solutions Enabler is available in both 32 and 64 bit support on some operating systems. The following prompt only displays if Solutions Enabler can support both 32 and 64 bit versions of the libraries on the host. A [Y]es response installs the 32 *and* 64 bit libraries. A [N]o response *only* installs the 32 bit libraries.

```
Install Solutions Enabler 64-bit Shared libraries ? [N]
```

Install the 32 bit Solutions Enabler libraries (/usr/symcli/shlib) and DBMS client libraries for all database daemons, except storora64d.

3. At the following prompt, specify whether to install the non-MT shared libraries. You should install these libraries if you plan on using the Device Masking compatibility binaries: fpath, fcompare, vcmfind, and fconfig.

Install NON MT Shared Libraries ? [N]

A [**Y**]es response installs the MT *and* the NON-MT libraries.

A [**N**]o response *only* installs the MT libraries.

4. At the following prompt, specify whether to *only* install the Solutions Enabler libraries that allow for the monitoring of statistics:

Install Shared Libraries That Will ONLY Allow
Monitoring Capability ? [N]

A [**Y**]es response *only* installs the libraries that enable Solutions Enabler's monitoring functionality. A [**N**]o response installs Solutions Enabler's full functionality, including monitoring and control.

Selecting to only install the monitoring functionality (**Yes**) will exclude the following from the installation:

- Control binaries, including symbcv, symsnap, symclone, symchg, symacl, symmask, symmaskdb, symmir, symreplicate, symrdf, symoptmz, symqos, symconfigure, symchksum, symrslv, symvg, symlv, symioctl, symrdb, symhost, and symhostfs
 - Database support (daemons and libraries)
 - Associated man pages
-

5. At the following prompt, specify whether to install the collection of binaries known as SYMCLI. A [**Y**]es response installs the SYMCLI binaries.

Install Symmetrix Command Line Interface (SYMCLI) ? [Y]

6. If you are installing Solutions Enabler on a host with a Tru64 UNIX, HP-UX, SunOS, or AIX operating system, the following prompt displays, asking whether to install one or more *optional* database components:

```
Install Solutions Enabler SRM Database Run Time
Components ? [N]
```

A [Y]es response displays the following prompts, depending on your operating system:

- SRM Oracle Database Component ?

This prompt only displays on operating systems where Solutions Enabler supports Oracle. A [Y]es response installs the *optional* Oracle daemon.

- SRM Informix Database Component ?

This prompt only displays on operating systems where Solutions Enabler supports Informix. A [Y]es response installs the *optional* Informix daemon.

- SRM Sybase Database Component ?

This prompt only displays on operating systems where Solutions Enabler supports Sybase. A [Y]es response installs the *optional* Sybase daemon.

- IBM UDB Database Component ?

This prompt only displays on operating systems where Solutions Enabler supports UDB. A [Y]es response installs the *optional* UDB daemon.

- SMI Provider and cimserver ?

A [Y]es response installs the Storage Management Initiative (SMI) Provider and cimserver. Together, these options provide the SMI-compliant Web-Based Enterprise Management (WBEM) interface for EMC Solutions Enabler. For more information, refer to *Configuring and Managing the SMI Provider* on page 3-39.

7. At the following prompt, specify whether to install the Solutions Enabler Java interface component. You should install this component if your Solutions Enabler application uses a Java interface. A [Y]es response installs the JNI component.

```
Install Option to Enable JNI Interface for EMC
Solutions Enabler APIs ? [N]
```

The installation program checks for sufficient disk space and unpacks the files related to the options you selected.

Once complete, the installation program will list EMC Solutions Enabler in the HAS BEEN INSTALLED list, as shown below. If you also have an existing installed base, additional text will display.

You may need to run the license management facility (symlmf).

Do not forget to run 'symcfg discover' if this is your first install or whenever your configuration changes.

```
#-----
# The following HAS BEEN INSTALLED in /opt/emc via the emc_install utility.
#-----
# ITEM  PRODUCT                                     VERSION
#-----
01      EMC Solutions Enabler                        V6.0.0

#-----
# The following is AVAILABLE TO BE INSTALLED from the selected Kit Location:
#-----
# ITEM  PRODUCT                                     VERSION
#-----
01      EMC Solutions Enabler                        V6.0.0

What would you like to do? Install (i) Uninstall (u) Register (r) Exit (x):
```

8. Enter **x** to exit the install script.

For instructions on uninstalling the Solutions Enabler kit, refer to *Uninstalling Solutions Enabler from UNIX* on page 4-4.

Step 6: Complete the Installation

This section explains how to complete your Solutions Enabler installation.

Cleanup Temporary Files

During installation, the install script writes out some temporary files to `/tmp`. In some cases these files will be removed when you reboot your system. If not, you may want to manually remove them to conserve disk space. Table 2-1 lists the temporary files.

Table 2-1 Installation Temporary Files

Filename	Purpose
<code>/tmp/emc_mount_path</code>	Holds the value that was entered for the <i>Kit Location</i> from the previous installation. This value is used as the default kit location in subsequent installs. For example: <code>EMC_MOUNT_PATH: /cdrom/UNIX</code>
<code>/tmp/emc_app_path</code>	Holds the value that was entered for the <i>Install root directory</i> from the previous installation. This value is used as the default <i>install root directory</i> in subsequent installs. For example: <code>EMC_APPLICATION_PATH: /opt/emc</code>

Unmount the CD

To unmount the CD, enter:

```
umount mount_point
```

Install the Kernel Patch

In a Linux on S390 installation, you must download and install a kernel patch. The patch and its documentation can be found on the the EMC ftp site:

```
ftp://ftp.emc.com/pub/elab/linux/dasd_SuSE_2.4.7.patch
```

```
ftp://ftp.emc.com/pub/elab/linux/  
ReadMe_dasd_SuSE_2.4.7.txt
```

Once you have installed the patch, you must insert the `s390ioctl.o` module into the kernel. Otherwise, the `symcfg discover` command will return error **40 No Devices Found**.

To insert the module into the kernel:

1. Issue the `lsmod` command to determine if the module is already inserted in the kernel.

If it is already inserted, there is no reason to complete this procedure. Otherwise, complete the remaining steps in this procedure.

2. Change directory to the following:

```
<kit location>/sdk/esl/bin
```

3. Issue the following command:

```
insmod s390ioctl.o
```

To remove the module from the kernel, issue the command:

```
rmmod s390ioctl
```

The `.o` extension is required on the `insmod` command, but cannot be present on the `rmmod` command.

Enable the Solutions Enabler Components

You must now enable your Solutions Enabler features by entering the appropriate license keys.

For instructions, refer to *Enabling Your Software* on page 3-2.

Installing Solutions Enabler on Windows

This section describes how to install Solutions Enabler in a Windows environment. The following procedure can be used for either a new installation, or to upgrade an existing installation.

Before starting this procedure, be sure to review Chapter 1 of this document and the *EMC Solutions Enabler Release Notes*.

To install Solutions Enabler:

1. Save all files and exit all Windows applications.
2. Insert the CD into the CD-ROM drive.

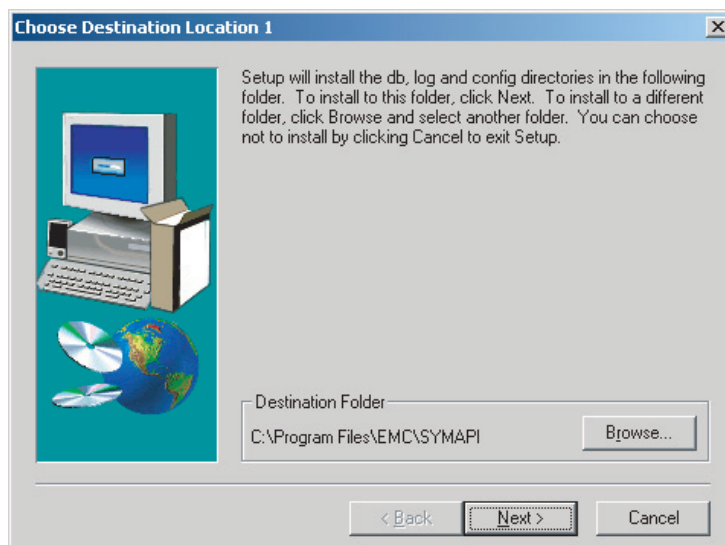
If autorun is enabled, the installation starts automatically. If autorun is not enabled, run `<CD-ROM drive>:\wsk60rt.exe`.

The **Solutions Enabler SYMCLI RT SETUP** dialog box opens.

3. Click **Setup**.

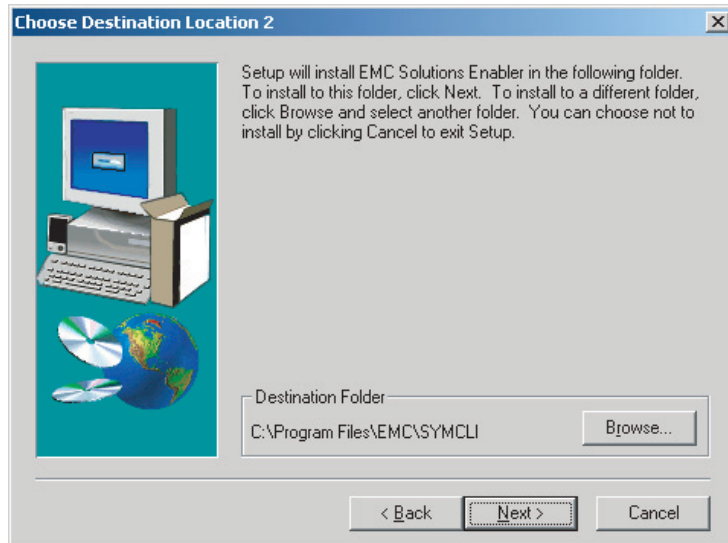
WinZip unzips the installation files and starts InstallShield.

The **Choose Destination Location 1** dialog box opens, prompting you to select an install directory for the `db`, `log`, and `config` directories.



4. Select an install directory and then click **Next**.

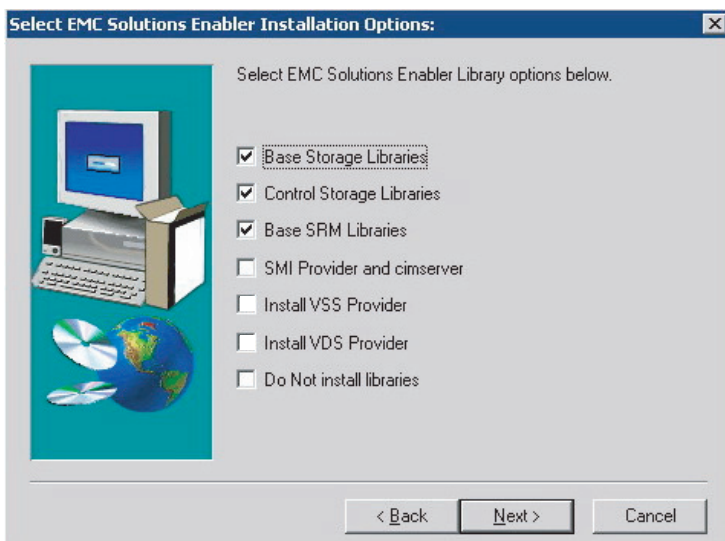
The **Choose Destination Location 2** dialog box opens, prompting you to select an install directory for Solutions Enabler.



5. Select an install directory and then click **Next**.

The **Select EMC Solutions Enabler Installation Options** dialog box opens, prompting you to select installation options.

The **VSS Provider** and **VDS Provider** options in the following example are only available/visible on Windows 2003 and .NET machines.

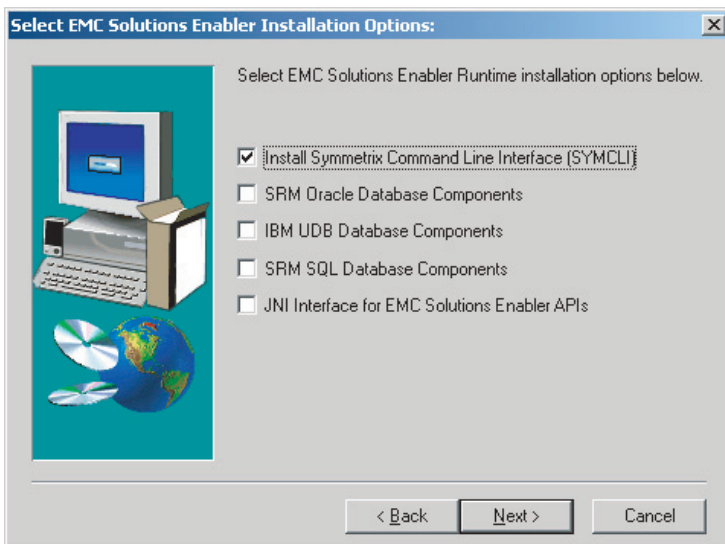


6. Select the installation options according to the following, and then click **Next**:
- To only install Solutions Enabler's core functionality, select **Do Not install libraries**. Solutions Enabler's core functionality includes `symapi`, `SymLvm`, `Storapi`, `Symapisrv`, `Storapid`, `Storcore`, `stordaemon`, and `storpds`.
 - To install additional functionality, select from the following:

Option	Description
Base Storage Libraries	Installs <code>StorSil</code> and <code>Storbase</code> . This option provides base storage and host specific functionality, and an interface to storage arrays for features like I/O scan, device listings, statistics, and showings.
Control Storage Libraries	Installs <code>StorSil</code> , <code>Storbase</code> , and <code>Storctrl</code> . This option provides the same functionality as the Base Storage option, but includes storage control functionality for features like Snap, device masking, and device mirroring.
Base SRM Libraries	Installs <code>Stormap</code> , which provides base Storage Resource Management functionality.
SMI Provider and cimserver	Installs the EMC Storage Management Initiative (SMI) Provider and the Pegasus CIM Server. Together, these options provide the SMI-compliant Web-Based Enterprise Management (WBEM) interface for Solutions Enabler. For more information, refer to <i>Configuring and Managing the SMI Provider</i> on page 3-39.
Install VSS Provider ^a	Installs the Microsoft Virtual Shadow Copy Service Provider.
Install VDS Provider ^a	Installs the Microsoft Virtual Disk Service Provider.

a. Only available/visible on Windows 2003 and .NET machines.

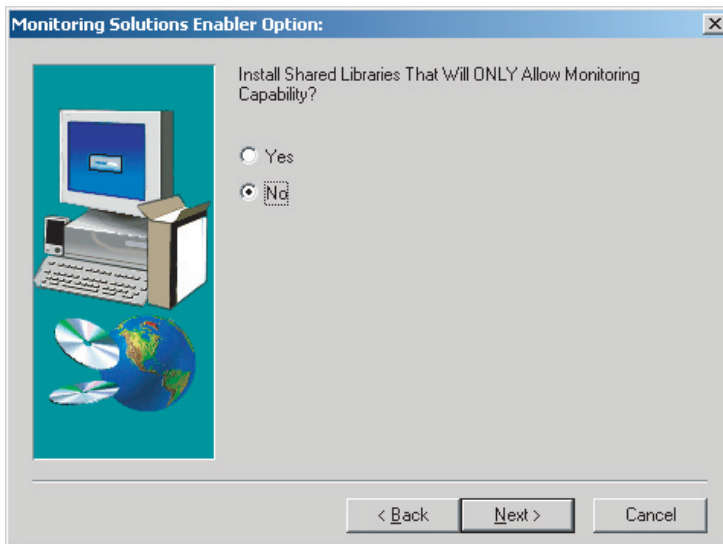
A second **Select EMC Solutions Enabler Installation Options** dialog box opens, prompting you to select additional options.



7. Select the installation options according to the following, and then click **Next**:

Option	Description
Install Symmetrix Command Line Interface (SYMCLI)	Installs the collection binaries known as SYMCLI.
SRM Oracle Database Components	Installs the <i>optional</i> Oracle daemon.
IBM UDB Database Components	Installs the <i>optional</i> UDB daemon.
SRM SQL Database Components	Installs the <i>optional</i> SQL daemon.
JNI Interface for Solutions Enabler APIs	Installs the Java interface component.

The **Monitoring Solutions Enabler Option** dialog box opens, prompting whether to *only* install Solutions Enabler's monitoring functionality.



8. Click **Yes** to *only* install Solutions Enabler's monitoring functionality, or **No** to install Solutions Enabler's full functionality, including monitoring and control.

Selecting to only install the monitoring functionality (**Yes**) will exclude the following from the installation:

- Control binaries, including symbcv, symsnap, symclone, symchg, symacl, symmask, symmaskdb, symmir, symreplicate, symrdf, symoptmz, symqos, symconfigure, symchksum, symrslv, symvg, symlv, symioctl, symrdb, symhost, and symhostfs
 - Database support (daemons and libraries)
 - Associated man pages
-

Click **Next** when done.

You are prompted to confirm your selection.

9. Click **Yes** to confirm your selection.

The installation program checks for sufficient disk space and unpacks the files related to the selected options.

If the Symmetrix Integration Utility (SIU) service is not already installed on the system, you are informed that the service is being installed and prompted to start it. The SIU service provides additional Windows disk management functions for working with TimeFinder® and SRDF® objects.

For more information on the SIU service, refer to the *EMC TimeFinder/Integration Modules Product Guide*.

10. Click **Yes** to have the installation program start the SIU service for you, or **No** to manually start it later.

You are prompted to install and start the SYMAPI server. The SYMAPI server allows the machine to act as the server for clients on other hosts.

For more information on the SYMAPI server, refer to *Client or Server Installation* on page 1-4.

11. Click **Yes** to have the installation program install/start the SYMAPI server for you, or **No** to manually install/start it later.

For more information on manually installing and starting the SYMAPI server, refer to the `symapisrv` command description in the *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide*.

You are prompted to install and start the storapid server.

For more information on the storapid server, refer to *Setting Up Daemons for Distributed Application Support* on page 3-30.

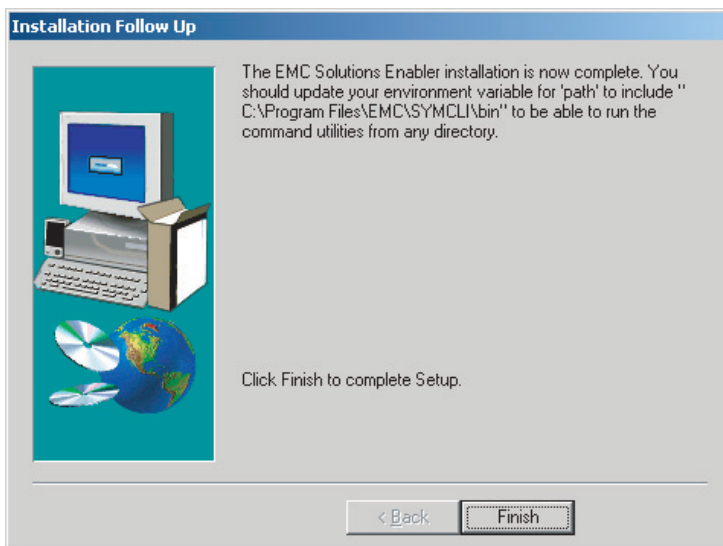
12. Click **Yes** to have the installation program start the storapid server for you, or **No** to manually start it later.

If you are installing the SMI Provider cimserver option, you are prompted to install and start the cimserver. Otherwise, the **Installation Follow Up** dialog box opens, as described on the next page.

13. Click **Yes** to have the installation program start the cimserver for you, or **No** to manually start it later.

For more information on manually starting the cimserver, refer to *Starting and Stopping the Pegasus CIM Server* on page 3-39.

The **Installation Follow Up** dialog box opens, recommending that you update the environment variable for `path` to include `C:\Program Files\EMC\SYMCLI\bin`. Updating this variable will allow you to run the command utilities from any directory.



14. Click **Finish** to complete the setup and then update the variable for `path`. Go to *Enabling Your Software* on page 3-2 when done.

Installing Solutions Enabler on OpenVMS

This section describes how to install Solutions Enabler in an OpenVMS environment. The following procedure can be used for either a new installation, or to upgrade an existing installation.

Before starting this procedure, be sure to review Chapter 1 of this document and the *EMC Solutions Enabler Release Notes*.

To install Solutions Enabler:

1. Insert the CD into the CD-ROM drive, and then mount it by entering:

```
mount/media=cdrom/undefined=(fixed:cr:32256)/over=id dkd600
```

2. Copy file `wsk60rt.sav` from the Solutions Enabler directory `[other.ovms]` to a temporary directory on your machine by entering:

```
copy /log dkd600:[other.ovms]wsk60rt.sav sys$sysdevice:[EMC.KITS]*.*
```

3. Extract the command procedure after setting `[set DEF SYS$SYSDEVICE:[EMC.KITS]` by entering:
`backup/select=instcli.com wsk60rt.sav/sav instcli.com;`
4. With both files (`instcli.com` and `wsk60rt.sav`) in the same temporary directory, run the installation procedure by entering:

```
@instcli.com
```

The installation produces a DCL command procedure (`EMC_CLI.COM`) to be called by the system `login.com` or by each user's login procedure.

5. Ensure that each SYMCLI user's login procedure calls the `EMC_CLI.COM` procedure to establish their proper SYMCLI environment.
6. Each user must have the following privileges for the SYMCLI to properly function. Take care when granting these privileges.
`DIAGNOSE` — Can diagnose devices.
`PHY_IO` — Can perform physical I/O.
`SHMEM` — Can create/delete objects in shared memory.
`SYSLCK` — Can lock system wide resources.
7. The installation is complete. Go to *Enabling Your Software* on page 3-2.

Once you have installed Solutions Enabler, you need to perform certain follow-up procedures to enable your software's features and to establish your command environment. This chapter provides the follow-up procedures for a Solutions Enabler installation.

- ◆ Enabling Your Software 3-2
- ◆ Initial Steps for Post Solutions Enabler Install..... 3-5
- ◆ Setting the CLI Path..... 3-6
- ◆ Setting the Online Help Path..... 3-7
- ◆ Symmetrix Gatekeeper Devices 3-8
- ◆ Database and Gatekeeper Locking..... 3-11
- ◆ Avoidance and Selection Files 3-14
- ◆ Options File..... 3-16
- ◆ SYMCLI Through a Remote Server 3-19
- ◆ Oracle Multiple Instances through Remote Server 3-28
- ◆ Setting Up Daemons for Distributed Application Support 3-30
- ◆ Configuring and Managing the SMI Provider..... 3-39

Enabling Your Software

Before you can use your installed Solutions Enabler components, you must first enable each component's functionality in the API by entering the appropriate license keys.

License Keys

Table 3-1 lists the possible Solutions Enabler keys:

Table 3-1 Solutions Enabler License Keys

Key Name	Description
BASE	Enables the Base component
Configuration Mgr	Enables the Symmetrix Configuration component
DeltaMark	Enables the DeltaMark facility
DevMasking	Enables the Device Masking component
Double Checksum	Enables the Double Checksum Oracle PAK commands
Open Replicator/DM for Symmetrix	Enables the Open Replicator/Data Mobility component
Open Replicator/LM for Symmetrix	Enables the Open Replicator/Live Migrator component
Optimizer	Enables the Symmetrix Optimizer control component
SRDF/A	Enables the SRDF/Asynchronous component
SRDF/Automated Replication	Enables the Automated Replication component
SRDF/Cluster Enabler for MSCS	Enables the Cluster Enabler component
SRDF/Consistency Groups	Enables the Consistency Group component
SRDF/Data Mobility	Enables the Data Mobility component
SRDF/Synchronous or SRDF	Enables the SRDF component
SRM/Mapping	Enables the Storage Resource Management component
SYMAPI Server	Enables the Remote SYMAPI Server (network) operations

Table 3-1 Solutions Enabler License Keys (continued)

Key Name	Description
SymmEnabler for Backup	Enables multiple Solutions Enabler components with a single key
Symmetrix Secure Erase	Enables the Symmetrix Secure Erase component
TimeFinder/Clone	Enables the TimeFinder/Clone component
TimeFinder/Consistency Groups	Enables Enginuity Consistency Assist (ECA)
TimeFinder/Exchange Integration Module	Enables the TEIM
TimeFinder/Mirror or TimeFinder	Enables the BCV control component
TimeFinder/Snap or EMC SNAP	Enables the TimeFinder/Snap component
TimeFinder/SQL Integration Module	Enables the TSIM

Enabling Components

To enable the components, do the following:

1. Ensure you are still at the install path by entering the following according to your operating system:

For	Enter
UNIX	/usr/storapi/bin
Windows	C:\Program Files\EMC\SYMCLI\bin
OpenVMS	SYMCLI\$BIN

On AS/400, add SYMCLI to your library list.

2. Invoke the Solutions Enabler License Management Facility (LMF) by entering the following:

```
symlmf
```

The LMF displays the following :

```

      E M C      S O L U T I O N S      E N A B L E R

SOLUTIONS ENABLER LICENSE MANAGEMENT FACILITY

```

3. At the following prompt, enter **y** to continue the registration:

```
Register Solutions Enabler LICENSE Key (y/[n]) ? y
```

4. At the following prompt, enter the license key of the component you want to enable:

```
Enter Solutions Enabler License Key:
```

LMF displays a message indicating success. For example, the following display indicates that the Base component was registered successfully:

```
The Solutions Enabler License Key for the BASE feature  
was successfully registered.
```

This message does not display in an OpenVMS environment.

5. At the following prompt, enter **y** to register another component, or **n** to quit:

```
Register Solutions Enabler License Key (y/[n]) ? y
```

6. Verify that the registered components are in the license file. The location of the license file varies according to the operating system:

Operating System	License File Location
UNIX	/var/symapi/config/symapi_licenses.dat
Windows	C:\Program Files\EMC\Symapi\config\symapi_licenses.dat
OpenVMS	SYMAPI\$CONFIG:SYMAPI_LICENSES.DAT

Initial Steps for Post Solutions Enabler Install

This section describes the initial steps you must consider before you begin using Solutions Enabler SYMCLI commands.

Build the SYMAPI Database

Before using the SYMCLI commands, you need to run the `symcfg discover` command to build your configuration (SYMAPI) database. This needs to be done once after installation, and after any changes are made to your Symmetrix configuration.

Set Environment Variables

Once you have completed the Solutions Enabler installation, you should set the environment variables or paths so you can directly access both the SYMCLI commands and the online help (man pages). The online help path allows you direct access to descriptions (man pages) of the command set. In addition, you can read the hard copy release notes, which describes pertinent information about this release including features, restrictions, and supported platforms.

Set Access Permissions to Directories

On UNIX platforms, the completed Solutions Enabler installation sets the access permissions on the Solutions Enabler directories to 755, which disables write access to other users beyond the owner. If you desire a different permission scheme, you can change it now.

Starting the SCSI Generic Driver

Linux requires that the SCSI generic driver be running. You can either compile it into the kernel or compile it as a loadable kernel module.

For instructions, refer to the `README` file in the top level directory of your Linux source package.

Setting the CLI Path

Before using SYMCLI, append the SYMCLI binary directories to your PATH environment variable according to your operating system.

UNIX

For UNIX C shell, ensure the following SYMCLI directories are appended to variable PATH:

```
set path = ($path /usr/storapi/bin;usr/storapi/storbin)
```

For UNIX Korn shell, ensure the following SYMCLI directories are appended to variable PATH:

```
export PATH=$PATH:/usr/storapi/bin;usr/storapi/storbin
```

For UNIX Bourne shell, ensure the following SYMCLI directories are appended to variable PATH:

```
PATH=$PATH:/usr/storapi/bin;usr/storapi/storbin
export PATH
```

Windows

For Windows, ensure the following SYMCLI directories are appended to the MS-DOS variable path:

```
C:\Program Files\EMC\SYMCLI\bin;C:\Program Files\EMC\SYMCLI\storbin
```

To append the variable path:

1. Do the following according to your operating system:
 - For *Windows 2000 Server* and *Windows 2000 Advance Server*, right-click **My Computer** and select **Management, System Information, Software Environment, Environment Variables**.
 - For *Windows 2000 Professional* and *Windows 2003*, right-click **My Computer** and select **Properties, Advanced, Environment Variables**.
2. Append the following to the value of **Path**:

```
;C:\Program Files\EMC\SYMCLI\bin;C:\Program Files\EMC\SYMCLI\storbin
```

OpenVMS

For OpenVMS, ensure the following SYMCLI directory has been defined for all users (use `emc_cli.com` in the system login.com):

```
SHOW LOGICAL SYMCLI$BIN
```

Setting the Online Help Path

A complete set of online help (man pages) is provided for SYMCLI. To access these man pages in your environment, do the following according to your operating system.

UNIX For UNIX C shell, ensure the following man page directories are added to variable MANPATH:

```
setenv MANPATH ${MANPATH}:/usr/storapi/man;/usr/storapi/storman
```

For UNIX Korn or Bourne shell, ensure the following man page directories are added to variable MANPATH:

```
export MANPATH=$MANPATH:/usr/storapi/man;/usr/storapi/storman
```

Windows For Windows, the manual pages are located, by default, in the following directory:

```
C:\Program Files\EMC\SYMCLI\man
```

To open a file, double-click it and select **NotePad** from the **Open With** dialog box.

OpenVMS For OpenVMS, help pages can be viewed with the DCL utility SYMHELP.

Symmetrix Gatekeeper Devices

Low level I/O commands executed using SYMCLI are routed to Symmetrix by way of a Symmetrix storage device that is specified as a *gatekeeper*. The gatekeeper device allows SYMCLI commands to retrieve configuration and status information from the Symmetrix array without interfering with normal Symmetrix operations. A gatekeeper is not intended to store data and is usually configured as a small device (under 10 MB). The gatekeeper must be accessible from the host where the commands are being executed.

CLARiiON storage systems do not use gatekeepers.

Choosing a Gatekeeper

During the installation of a Symmetrix array, the EMC Customer Engineer ensures that a certain number of gatekeepers are assigned. If at some point, a host issuing commands is unable to communicate with a designated gatekeeper, SYMCLI automatically chooses (by default) one of the available devices to function as the gatekeeper.

The `syminq` command will identify the devices that EMC has configured as gatekeepers. The following is sample output for this command. EMC configured gatekeepers display a **GK** in the type field.

Device			Product		Device	
Name	Type	Vendor	ID	Rev	Ser Num	Cap (KB)
\\.\PHYSICALDRIVE0		COMPAQPC	DDYS-T09170N	B93E	N/A	N/A
\\.\PHYSICALDRIVE1		DEC	HSG80	V86P	N/A	N/A
\\.\PHYSICALDRIVE2		DEC	HSG80	V86P	N/A	N/A
\\.\PHYSICALDRIVE3		EMC	SYMMETRIX	5568	02000000	7680
\\.\PHYSICALDRIVE4	GK	EMC	SYMMETRIX	5568	02720008	4800
\\.\PHYSICALDRIVE5	GK	EMC	SYMMETRIX	5568	0271B008	4800
\\.\PHYSICALDRIVE6	GK	EMC	SYMMETRIX	5568	0271C008	4800
\\.\PHYSICALDRIVE7	GK	EMC	SYMMETRIX	5568	0271D008	4800
\\.\PHYSICALDRIVE8	GK	EMC	SYMMETRIX	5568	0271E008	4800
\\.\PHYSICALDRIVE9	GK	EMC	SYMMETRIX	5568	0271F008	4800
\\.\PHYSICALDRIVE10		EMC	SYMMETRIX	5568	00000000	7680
\\.\PHYSICALDRIVE11		EMC	SYMMETRIX	5567	40000000	7680

To explicitly designate one or more gatekeepers, use the `symgate` command to add them to the gatekeeper list within the config database. In the gatekeeper list, you can associate (dedicate) a device as the gatekeeper to a device group. SYMCLI routes low level I/O commands to devices in the device group by way of the assigned gatekeeper.

To display a list of designated gatekeepers that exist in your config database, use the SYMCLI `symgate` command.

If you define a gatekeeper but do not associate it to the device group, SYMCLI chooses the first gatekeeper on the list. If that device is not accessible or is currently in use, SYMCLI will try using other gatekeeper devices in the gatekeeper list. If you do not define a gatekeeper, SYMCLI selects gatekeepers according to a dynamic selection algorithm. It is recommended to have a designated gatekeeper.

Using the `gkavoid` and `gkselect` Files

The optional `gkavoid` and `gkselect` files affect calls to various online type SYMCLI commands that use a gatekeeper to communicate with a Symmetrix array.

`gkavoid` affects calls to various online type SYMCLI commands which use a gatekeeper to communicate to a Symmetrix array. A gatekeeper whose `PdevName` matches any of the entries specified in the `gkavoid` file, will not be chosen as a gatekeeper to communicate with the Symmetrix. This could be useful to designate certain Symmetrix devices that should not be used as gatekeepers. The gatekeeper avoidance file is formatted with physical device names with one `PdevName` (`/dev/rdisk/c2t0d1s2`) per line.

The gatekeeper selection file (`gkselect`) is formatted with physical device names, with one `PdevName` (for example, `/dev/rdisk/c2t0d1s2`) per line. Those devices whose `PdevNames` match any of the entries specified in the `gkselect` file will be the only devices used as gatekeepers. This can be useful if you want to only use specific devices as gatekeepers.

Creation of the `gkselect` file is optional, however, doing so will override any gatekeepers defined by the `symgate` command.

If a device is listed in both the `gkavoid` file and the `gkselect` file, the device will be avoided.

Using a Dedicated Gatekeeper

A gatekeeper device can be *dedicated* or not. If a gatekeeper is dedicated (defined), its Symmetrix device should not be used by the host system for normal data processing.

Sizing a Gatekeeper

When a Symmetrix array is installed, the EMC Customer Engineer selects and configures Symmetrix devices with less than 10 cylinders (less than 5 MB) for use as gatekeeper devices.

However, the gatekeeper device must be at least as large as the minimum volume size accessible by your host, which is usually, 6 cylinders, 2.8 MB. Consult your host documentation for the minimum device size accessible by your particular host to determine the minimum gatekeeper device size for your environment.

You can determine the storage size of a Symmetrix device using:

- ◆ The `sympd` command using the `list` and `show` arguments as follows:
 - `list` — displays a list of physical device names and storage size (in MBs) for a specific Symmetrix.
 - `show` — displays the parameters of a specified physical device that includes the device capacity or size in blocks and megabytes.
- ◆ The `syminq` command and specifying the physical device name.

Usually the EMC Customer Service Engineer configures a few Symmetrix devices for use as gatekeepers. You can distinguish these devices in a list executed by `syminq [PdevName]` by locating a symbol GK next to the PdevName (physical device name). Otherwise, they are not easily distinguished from other devices in other configuration lists, except perhaps by their size, which tends to be smaller than other devices.

For more information about gatekeepers, refer to the *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide*.

Database and Gatekeeper Locking

Within a SYMCLI session, gatekeeper and database locks are used to avoid conflicts in accessing a Symmetrix array by way of gatekeepers or the configuration database.

CLARiiON storage systems do not use gatekeepers.

Semaphore Requirements on UNIX

On a UNIX system, SYMCLI allocates a system semaphore for each accessed Symmetrix gatekeeper device and SYMAPI database file. These semaphores are not deallocated from the system, but are re-used whenever these resources are accessed again.

An adequate number of semaphores should be configured into the UNIX kernel to meet the SYMCLI semaphore requirements as follows:

- ◆ One semaphore ID for each Symmetrix gatekeeper device.
- ◆ One semaphore ID for each Symmetrix host database file in use.
The number of system-wide semaphores is specified by the UNIX kernel parameter `semmns`, or its equivalent.
- ◆ A minimum of three semaphores per semaphore set.
The maximum number of semaphores per semaphore set is specified by the UNIX kernel parameter `semmns1`, or its equivalent.
- ◆ A minimum of three operations per `semop` call.
The maximum number of operations per `semop` call is specified by the parameter `semopn`, or its equivalent.

These requirements are usually within the bounds of the default semaphore parameter settings on a UNIX system. However, for information about maximizing these parameters on your specific platform, see *Host Specific Issues* on page A-1.

Meeting Semaphore Requirements

If the requirements are not within the bounds of the default semaphore parameter settings on a UNIX system, the UNIX kernel must be reconfigured. If the UNIX kernel is not reconfigured, the SYMCLI gatekeeper and database locking will fail. For more information about adjusting semaphore parameters for your operating system, refer to *Host Specific Issues* on page A-1.

Refreshing the Semaphores

After you have reconfigured the UNIX kernel, you may need to reboot the UNIX system to refresh the kernel semaphore structures.

You can use the following UNIX command to view the currently allocated system semaphores:

```
ipcs -s
```

De-Allocating Semaphores

If you exceed the maximum number of semaphores allocated, you may need to de-allocate system semaphores in order to obtain more semaphores.

To de-allocate a system semaphore, use the following UNIX command:

```
ipcrm -s IpCID
```

Semaphore Identifier

SYMCLI uses the UNIX function `ftok()` to derive a semaphore identifier from the gatekeeper or database pathname. The `ftok()` function generates a unique identifier based on the i-node number of the gatekeeper raw device name or database pathname.

The SYMCLI semaphore lock functions depend on the i-node remaining constant during the course of an operation in order to acquire and release a specific lock. In the case of a database lock, if the file is superseded, the i-node will be changed and the lock functions will be unable to release the associated semaphore.

OpenVMS Locking

On OpenVMS, SYMCLI uses the Distributed Lock Manager to accomplish locking. These locks are automatically de-allocated from the system when the last process, which has opened the lock, finishes or is terminated. There is no kernel configuration requirement. The lock name is derived from the gatekeeper or database pathname.

Windows Locking

On Windows, SYMCLI allocates named mutexes to accomplish locking. These mutexes are automatically de-allocated from the system when the last thread which has opened the mutex finishes accessing the mutex, or is terminated. There is no mutex kernel configuration requirement. The mutex name is derived from the gatekeeper or database pathname.

Avoidance and Selection Files

The following files can exist in the SYMAPI configuration directory, and limit the scope or change the performance of SYMCLI online commands, particularly, `symcfg discover` and `syminq`:

- ◆ `gkavoid`
- ◆ `gkselect`
- ◆ `inqfile`
- ◆ `symavoid`

These files and the following text are for experienced SYMCLI or SYMAPI users and are not a prerequisite for normal use.

These files can be used to customize and streamline command line coding to your specific environment. The files are located in the SYMAPI configuration directory:

<code>/var/symapi/config</code>	on UNIX systems
<code>your_install_directory\config</code>	on OpenVMS and Windows systems

Editing and File Format

These are editable files with device names or Symmetrix IDs you can use to limit SYMCLI or SYMAPI from seeing certain Symmetrix units, devices, or gatekeepers which would otherwise be affected by various commands.

The files hold either physical device names (*PdevNames*) or Symmetrix IDs (*Symmids*) with line entries having only one device name or ID per line. Lines beginning with a “#” (comment) are ignored by SYMCLI.

gkavoid and gkselect

The optional `gkavoid` and `gkselect` files affect calls to various online type SYMCLI commands that use a gatekeeper to communicate with a Symmetrix array.

For more information on these files, refer to *Symmetrix Gatekeeper Devices* on page 3-8.

inqfile

The `inqfile` file affects calls to `syminq` and `symcfg discover`, which will find only the PdevNames specified in this file. This may be useful if you want to limit the command(s) to affect only certain Symmetrix devices from your host. The inquiry file is formatted with physical (host) device names with one PdevName (`/dev/rdisk/c2t0d2s2`) per line.

symavoid

The `symavoid` file affects the operation of `symcfg discover` so that it skips over looking for devices that belong to the Symmetrix units identified in this file. This may be useful if there are multiple Symmetrix units connected to the host, that you want SYMCLI to avoid. The Symmetrix avoidance file is formatted with 10- to 12-character Symmetrix IDs with one ID per line.

To obtain a list of Symmetrix IDs, use the following command:

```
syminq -symmids
```

Options File

The `options` file in the SYMAPI configuration directory contains default constants that can be set to change defaults to certain options within various SYMCLI commands and associated SYMAPI calls. It can be used to customize and streamline command line coding to your specific environment.

This file and the following text are for experienced SYMCLI or SYMAPI users and are not a prerequisite for normal use.

The `options` file must be created and placed in the SYMAPI configuration directory:

<code>/var/symapi/config</code>	on UNIX systems
<code>C:\Program files\EMC\Symapi\config</code>	on Windows systems
<code>your_specific_installation_directory</code>	on OpenVMS, AS/400, and OS/390 systems

Editing the Options File

Once this file is created, you can edit it to change the default behavior of certain SYMCLI or SYMAPI command options. The file contains editable parameters to set certain optional defaults in the line entries. SYMAPI ignores lines beginning with a “#” (comment).

Table 3-2 lists the possible parameter entries for this file:

Table 3-2 Optional Behavior Parameters for Options File^a

Optional Behavior Parameter	Component Affected
SYMAPI_ACC_ADMIN_VIA_SERVER	Access Control
SYMAPI_ACC_DISPLAY_VIA_SERVER	Access Control
SYMAPI_ALLOW_RDF_SYMFORCE	SRDF
SYMAPI_ALLOW_SCRIPTS_VIA_SERVER	TimeFinder
SYMAPI_APPREG_AUTO_EXPIRATION	Base
SYMAPI_APPREG_EXPIRATION_PERIOD	Base
SYMAPI_BCV_SINGULA_INTERNAL	TimeFinder

Table 3-2 Optional Behavior Parameters for Options File^a (continued)

Optional Behavior Parameter	Component Affected
SYMAPI_CG_TIMEOUT	Consistency Groups
SYMAPI_CG_TIMEOUT_ACTION	Consistency Groups
SYMAPI_COLLAPSE_STRIPED_META_EXTENTS	Mapping
SYMAPI_CTRL_OF_NONVISIBLE_DEVS	Base, (all)
SYMAPI_CTRL_VIA_SERVER	Base, (all)
SYMAPI_DATED_LOGFILE_NAME	Base, (all)
SYMAPI_DB_FILE_COMPRESSION	Base, (all)
SYMAPI_DEFAULT_BCV_ESTABLISH_TYPE	TimeFinder
SYMAPI_DEFAULT_BCV_RESTORE_TYPE	TimeFinder
SYMAPI_DEFAULT_BCV_SPLIT_TYPE	TimeFinder
SYMAPI_DEFAULT_SNAP_TERM_TYPE	TimeFinder
SYMAPI_IO_DRAIN_TIMEOUT	Mapping, TimeFinder
SYMAPI_IO_THAW_INTERVAL	Mapping, TimeFinder
SYMAPI_LOGFILE_DATE_FORMAT	Base, (all)
SYMAPI_LOGFILE_RETENTION	Base, (all)
SYMAPI_MAX_CLIENTS	Base
SYMAPI_PARALLEL_RA_GROUPS	SRDF
SYMAPI_RDF_RW_DISABLE_R2	SRDF
SYMAPI_SERVER_SECURITY_LEVEL	Base, (all)
SYMAPI_SNAP_CONTROL_INTERVAL	TimeFinder
SYMAPI_SYNC_DIRECTION	SRDF, TimeFinder, TimeFinder/Snap
SYMAPI_TF_COUNT_MODIFIED_TRACKS	TimeFinder
SYMAPI_THREE_CHAR_SYMDEVNAME	Base, (all)

Table 3-2 Optional Behavior Parameters for Options File^a (continued)

Optional Behavior Parameter	Component Affected
SYMAPI_WAIT_FOR_BCV_BG_SPLIT	TimeFinder
SYMAPI_WAIT_FOR_BCV_SYNC	TimeFinder
SYMAPI_WAIT_ON_LOCKED_GK	Base, (all)

a. For more information on these optional parameters, refer to the *Options File* description in the *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide*.

Removing Default Options

To remove a default option, remove the line entry, rename the file, or comment the line by adding a pound (#) sign at the beginning of the line entry.

SYMCLI Through a Remote Server

If you intend to use the remote SYMAPI server feature for your SYMCLI interface, you need to be aware of the following files:

- ◆ `netcnfg` on the client side lists SYMAPI server services on any host, local or remote, to which the SYMCLI can connect.

To execute a remote SYMCLI session across a TCP/IP network connection, the client `netcnfg` file must map to a running SYMAPI server.

`netcnfg` on the server side lists named SYMAPI server services that can be used to start the SYMAPI server using the `-service` option. The `netcnfg` file is optional on the server host.

- ◆ `symapinlck` and `symapislck` are internal files used by the server as a lock file to guarantee a single port (2707 is the default) listener for a network service. Do not delete or alter these files.
- ◆ `nethost` lists the nodes/users allowed to connect to the server to execute remote SYMAPI functions.
- ◆ `symapisrv_trust.pem` and `symapisrv_cert.pem` are certificate files used by the client to verify the identity of the server for remote sessions.

Securing Remote Access

By default, the SYMAPI client and server are initially configured to negotiate either secure or non-secure sessions. To modify this default behavior, you can configure the security level at which the client and server are operating.

When configuring the security level, it is important to know that the security level specifies the capability of the local side and the local side's expectation of the remote side. In addition, it is important to know whether the host is SSL-capable or SSL-incapable. For information on whether a host is SSL-capable, refer to the *EMC Solutions Enabler V6.0 Support Matrix*.

The possible security levels are:

- ◆ **Level 3 (SECURE)** — Indicates that only secure sessions will be negotiated between the client and server. This is the highest level of security and should only be used when there is no chance of an SSL-incapable client attempting to connect with the server, or a new client connecting to an SSL-incapable server.
- ◆ **Level 2 (ANY)** — Indicates that either secure or non-secure sessions will be negotiated between the client and server. This is the default mode, allowing communication between the client and server where the SSL capabilities do not agree, and between supported and non-supported platforms.
- ◆ **Level 1 (NON-SECURE)** — Indicates that only non-secure sessions will be negotiated between the client and server. This level is intended as a last resort in situations where SSL cannot be used for some reason, or is undesirable. In addition, this level can also be useful in matters of performance and availability.

Configuring the Client

To configure the client's security level, use either of the following methods:

The following methods are listed in order of precedence to the client.

- ◆ Define the client's security level in the `netcnfg` file. For instructions on editing the `netcnfg` file, refer to *Editing the netcnfg File* on page 3-23.
- ◆ Define the client's security level in the `options` file. This file provides the global security level when no other source (`netcnfg` file) specifies a more specific level. For instructions on editing the `options` file, refer to *Editing the Options File* on page 3-16.

Configuring the Server

To configure the server's security level, use one of the following methods:

The following methods are listed in order of precedence to the server.

- ◆ Specify a security level (`-seclevel`) while starting the server with the `symapisrv` command. For instructions, refer to *Starting the Server* on page 3-25.
- ◆ Define the server's security level in the `netcnfg` file in conjunction with the `-service` operand or the `symapisrv` command. For instructions on editing the `netcnfg` file, refer to *Editing the netcnfg File* on page 3-23.
- ◆ Define the server's security level in the `options` file. This file provides the global security level when no other source (a `symapisrv` command or `netcnfg` file) specifies a more specific level. For instructions on editing the `options` file, refer to *Editing the Options File* on page 3-16.

Client/Server Behavior

If the client and server security levels are the same, implied or configured, the two will negotiate a session of the desired type.

Client Security Level	Server Security Level	Negotiated Session Type
SECURE	SECURE	SECURE
NONSECURE	NONSECURE	NONSECURE
ANY	ANY	SECURE

If the client and server security levels are different, either the client or server must be configured as ANY to negotiate the session.

Client Security Level	Server Security Level	Negotiated Session Type
SECURE	ANY	SECURE
ANY	SECURE	SECURE
NONSECURE	ANY	NONSECURE
ANY	NONSECURE	NONSECURE

Client/Server Compatibility

To secure a session, both the client and server must be at Solutions Enabler Version 6.0 and must be running on SSL-capable platform. The following table details what happens when a session is negotiated between mismatched client and server versions.

Client Version/Security Level	Server Version/Security Level	Session Results
V6.0/NONSECURE	V6.0/NONSECURE or ANY	Accepted
V6.0/SECURE	V6.0/SECURE or ANY	Accepted
V6.0/NONSECURE	V6.0/SECURE	Rejected by the server with return code SYMAPI_C_SECLEVEL_REMOTE_REFUSAL.
V6.0/SECURE	V6.0/NONSECURE	Rejected by the server with return code SYMAPI_C_SECLEVEL_REMOTE_REFUSAL.
V6.0/SECURE	Prior to V6.0	Rejected by the client with return code SYMAPI_C_SECLEVEL_LOCAL_REFUSAL.
V6.0/NONSECURE or ANY	Prior to V6.0	Accepted
Prior to V6.0	V6.0/SECURE	Rejected by the server with return code SYMAPI_C_SECLEVEL_REMOTE_REFUSAL. ^a
Prior to V6.0	V6.0/NONSECURE or ANY	Accepted

a. Older clients will see SYMAPI return code SYMAPI_C_HANDSHAKE_FAILED.

Certificate Files

The following certificate files enable the client and server to verify each other's identity:

- ◆ `symapisrv_cert.pem` is the SYMAPI server's certificate chain file. It is initially signed using the EMC SPEA Root certificate, and is made up of the certificate, associated private key, and the root certificate concatenated in one file. This file must be installed in the `config` directory on the SYMAPI server.

To install another certificate, you must overwrite the original file with the new file.

- ◆ `symapisrv_trust.pem` is the EMC SPEA Root certificate used to sign the SYMAPI server certificate file. It must be installed in the `config` directory on every client.

To install another certificate, you must either overwrite the original file, or append the new certificate to the original.

Before replacing a certificate file, you should save a copy of the original file.

Editing the netcnfg File

At this point in the install, the netcnfg file is a template and an editable file located in directory:

/var/symapi/config/	on UNIX systems
\Program Files\EMC\symapi\config\	on Windows systems
SYMAPI\$CONFIG:	on OpenVMS systems

Using a text editor, a System Administrator must add the network services to the file in the following format:

```
service_name domain_name network_protocol server_node_name server_network_address
port_number security_level
```

where:

service_name is the name of the service.

domain_name should be unspecified and substituted with a hyphen (-).

network_protocol must be TCP/IP.

server_node_name is the name of the server.

server_network_address is the network address of the server.

An unspecified **server_node_name** or **server_network_address** can be substituted with a hyphen (-). But at least one must be specified.

port_number is the server port number.

security_level is the type of connections the server is expecting to negotiate. Possible values are SECURE, ANY, and NONSECURE. In addition, you can specify a hyphen (-) to use the platform's default setting. For more information, refer to *Securing Remote Access* on page 3-19.

Example: In the following example, two site specific service names (SYMAPI_SERVER and BACKUP_SERVER) are specified as available by the administrator:

```
SYMAPI_SERVER - TCPIP node001 12.345.67.89 7777 ANY
BACKUP_SERVER - TCPIP node002 - 6666 SECURE
```

Comment text can be entered by placing a pound sign (#) in the first character space of the comment line.

Configuring the Trusted-User Host Access File

An optional file (`nethost`) for trusted-user host access can also be present in the server configuration directory. When this file exists (maintained by a System Administrator), only the nodes/users listed in this file are allowed to connect to the server to execute remote SYMAPI functions. The trusted host file uses the following format:

```
node          user-1 [, ..., user-n]
address       user-1 [, ..., user-n]
*             user-1 [, ..., user-n]
node          *
address       *
*             *
```

* denotes a wild card for any host or any user.

The server considers the contents of this file before deciding whether it will negotiate a secure session with the client. If the client host and user are not defined in the `nethost` file, a secure session will not be negotiated. For more information on the SSL security settings, refer to *Securing Remote Access* on page 3-19.

Registering the Server in the Windows Service Control Manager

For the Windows server platforms, you must register (install) the SYMAPI server option as a Windows service in the Service Control Manager. Using the command line of the MS-DOS window, you must add the SYMAPI server option to the **Services** list using the following SYMCLI command form:

```
symapisrv install
```

Typically, you do not need to specify any optional argument unless you have a particular network in mind.

For more information on this command, consult your Network Administrator and refer to the *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide*.

Once the SYMAPI server is installed as a service, it can be controlled with the Windows **Services** dialog box, which is accessible through the **Administrative Tools** application under **Control Panel**.

When the service is running, the `symapi.dll` cannot be replaced in the `\winnt\system32` directory.

Starting the Server

In UNIX and Linux environments, before you begin to use SYMCLI on the client, you must start the remote SYMAPI service using the following command executed from the server side:

```
symapisrv [-background]
          start
          [-service Servicename] |
          [-port Port]
          [-secllevel SecLevel]
```

Where `-secllevel` specifies the server's security level. Possible values are `SECURE`, `ANY`, and `NONSECURE`. This option cannot be used when starting the server with a service name. If starting the server with a service name, you must define its security level in either the `netcnfg` file or `options` file. For more information, refer to *Securing Remote Access* on page 3-19.

For OpenVMS, you can automate this process by setting the default to `SYMAPI$CONFIG` and entering `@SYMAPI$SRV.COM`.

Example: To start a SYMAPI server at the default port with the security level set to accept both secure and non-secure (legacy) sessions:

```
symapisrv -secllevel ANY start
```

At this point, you can use the SYMCLI to execute requests via the remote server.

Setting Environment Variables for Remote Access

To use SYMCLI through a remote SYMAPI service, you should set environment variable `SYMCLI_CONNECT` to an available service name of the server connection (defined in `netcnfg`). For example, for service name `SYMAPI_SERVER`, set the environment variable as follows:

```
setenv SYMCLI_CONNECT SYMAPI_SERVER for UNIX C shell
set SYMCLI_CONNECT=SYMAPI_SERVER for Windows
```

To determine what network services are available, (or to read the `netcnfg` file), enter:

```
symcfg list -service
```

Connection variable `SYMCLI_CONNECT_TYPE` should define the local/remote mode of the local host (client). Possible values for the client are:

`REMOTE`

Defines a client operation in which all the remote SYMCLI commands are strictly executed on the server, and the Symmetrix database is strictly read and updated remotely.

`REMOTE_CACHED`

Defines a client operation in which the remote database is modified remotely but cached in memory locally. Those functions that are control operations or which modify the database are executed remotely. These modifications to the remote database are then cached locally. This mode is not supported on the AS/400 platform.

`LOCAL`

Defines a local connection to the Symmetrix. (Not used for a client-server connection.)

For example, to set the connection environment variables for a locally-cached remote operation, enter:

```
setenv SYMCLI_CONNECT_TYPE REMOTE_CACHED
```

Stopping the Server

To stop the remote SYMAPI service from the client or server side, use the following command:

```
symapisrv  
    stop  
    [-service Servicename]  
    [-port Port]
```

Example: To stop the SYMCLI session via the remote server with service SYMAPI_SERVER, enter the following on your host or server:

```
symapisrv -serv SYMAPI_SERVER stop
```

Uninstalling the Server (Windows)

To uninstall the SYMAPI Server:

1. From MS-DOS, change directories to the SYMAPI server installation directory, by default:

```
C:\Program Files\EMC\SYMCLI\bin
```

For servers with an Itanium processor, the default installation directory is
C:\Program Files (x86)\EMC\SYMCLI\bin.

2. Stop the SYMAPI server, by entering the following:

```
symapisrv stop
```

3. Enter the following:

```
symapisrv uninstall
```

The SYMAPI Server is uninstalled and no longer appears in the list of services.

Oracle Multiple Instances through Remote Server

If you have the Storage Resource Management (SRM) license and intend to perform database mapping calls from your host to a remote server that has more than one Oracle instance, you must complete the following procedure:

1. With the remote SYMAPI service stopped, set the remote server UNIX environment variables `ORACLE_HOME` and `ORACLE_SID` for the system requirements. When set, re-start `symapisrv`.
2. Configure Oracle SQL*Net (V7) or Net8 to include other instance names (TNS names) in a network service. The TNS names are located in the `$ORACLE_HOME/network/admin/tnsnames.ora` file. The Oracle instance to which your `ORACLE_HOME` points to is the only instance that must have the TNS names registered.
3. Configure the Oracle listener service for the other Oracle instances with which you need to work.
4. Test your Oracle environment for a valid configuration by running `$ORACLE_HOME/bin/sqlplus` as follows:

```
sqlplus user/passwd@service
```

where:

user/passwd describes your Oracle user name and password

service is the TNS name you registered for the Oracle instance.

For more information about configuring SQL*Net or Net8, refer to the appropriate Oracle documentation.

5. Set the EMC environment variable `SYMCLI_RDB_CONNECT` to describe your user name, password, and service name with the format `usr/passwd@service` to the instance of choice.

Client/Server RDBMS Environment Variable Behavior

The commands `symioctl` and `symrdb` scan the client's current environment variables and apply them across the client/server connection. For example, when the following is invoked from the client:

```
symrdb -type oracle list
```

`symrdb` will search for `ORACLE_HOME` and `ORACLE_SID` on the client side. If found, the variables are passed to the SYMAPI server and used with subsequent database mapping calls.

Set the `LD_LIBRARY_PATH` environment variable for all databases except Oracle and SQL Server.

Setting Up Daemons for Distributed Application Support

To improve performance on a number of applications or scripts running at once, you can employ Solutions Enabler daemons (services) that run in the background with root privileges to a local Symmetrix storage resource. Applications do not have to run as a privileged user.

A base daemon (`storapi`) can be used to coordinate all Symmetrix locks and parallel application syscalls to your operating system kernel, which optimizes their operations (such as TimeFinder-type actions).

For storage resource management (SRM) applications, there are a number of vendor-specific database daemons available to improve the speed of database access or mapping operation. SRM database performance is improved by using a persistent database connection, a fast communication mechanism, and parallel operations. For SRM, a single database daemon can support connections to multiple instances/databases.

When your host is locally-connected to the Symmetrix array, applications and daemons must reside in that host. However, for client/server systems, the storage management applications reside in the client, but the daemons must reside in the SYMAPI server.

On Windows platforms, daemons run as services. After installing a service, it appears in the service list dialog box and can be accessed from the Windows **Services** control panel. The service will not restart at boot unless you modify the startup options through the service list dialog box.

Table 3-3 lists the possible Solutions Enabler daemons.

Table 3-3 Supported Solutions Enabler Daemons

Daemon	Function or Specific Application
<code>storgnsd</code>	Group Name Services support
<code>storifmxd</code>	Informix Database support
<code>storora64d</code>	Oracle Database 64-bit support
<code>stororad</code>	Oracle Database support
<code>storrdfd</code>	RDF consistency support

Table 3-3 Supported Solutions Enabler Daemons (continued)

Daemon	Function or Specific Application
storsqlld	SQL Server Database support
storudbd	DB/UDB support
storsybs11d	Sybase 11 Database support
storsybs12d	Sybase 12 Database support
storsybs12.5d	Sybase 12.5 Database support
storsybs12.5_64d	Sybase 12.5 (64-bit) Database support

You can start and stop a daemon at any time. If an application is already connected by way of the daemon and the daemon is stopped, the application continues to run (using the database engine binaries), but may experience degraded performance. If an application is already connected without the daemon and the daemon is started, the application automatically starts using the daemon.

Starting Daemons

To implement daemons for distributed application support, you must start each of the daemons you need by applying the following command:

```
stordaeon start DaemonName [-wait Seconds]
```

The `stordaeon` command requires a path of `/usr/storapi/storbin`.

By default, the `stordaeon` command waits 30 seconds to verify that the daemon is running. To override this, use the `-wait` option. For example, to start an SRM daemon for an Oracle database and wait five seconds for it to come up, enter:

```
stordaeon start stororad -wait 5
```

Stopping Daemons

To stop a daemon, apply the following command:

```
stordaeon shutdown DaemonName|all [-wait Seconds]
[-immediate]
```

Viewing Daemons

To view what daemons are present, enter either of the following:

```
stordaeomon list [-running] [-all] [-v]
```

or:

```
stordaeomon show
```

For the database daemons, an instance identifier is appended to the daemon name. For example, a `stororad` daemon started with the instance name `ords` would display as `stororadords`.

Configuring Daemons to Auto-Start on Boot

If you want daemons to automatically start upon reboot of your system, you can create an auto-start script. For information on creating auto-start scripts, refer to Appendix B.

Building a daemon_users File

The `storapid` daemon runs with root/administrator privileges, which enable it to handle the tasks required by SYMCLI commands (and any SYMAPI call) that require privileged access. This enables non-privileged users to run the SYMAPI application.

For example, when a SYMAPI call attempts to open a gatekeeper (which requires a privileged user), the request is actually passed to the `storapid` process, which will open the gatekeeper device. If you were to run `adb` and check the per-process file table, the open files would appear in the `storapid` process, not in the user process. From this point on, the transfer CDB requests are passed to `storapid` since it is the process that opened the gatekeeper.

By default, `storapid` only accepts connection requests from root and administrator type users. For non_root users to use this feature, you need to create a special file (`daemon_users`) with a list of allowed usernames.

The `daemon_users` file is an editable template file located in directory:

<code>/var/symapi/config/</code>	on UNIX systems
<code>\Program Files\EMC\symapi\config\</code>	on Windows systems
<code>SYMAPI\$CONFIG:</code>	on OpenVMS systems

Using a text editor, a System Administrator can add entries to this file using either of the following formats:

<code>smith storapid</code>	Local user smith is authorized to use the <code>storapid</code> daemon.
<code>\ENG\smith stororad</code>	Windows local user smith in the ENG domain is authorized to use the <code>storapid</code> daemon.
<code>smith storora*</code>	The * is a wildcard. Local user smith is authorized to use any daemon whose name begins with <code>storora</code> . For example, the SRM Oracle DB daemons.
<code>smith stororad freeze,...</code>	Local user smith is authorized to perform freeze and thaw operations via the <code>stororad</code> daemon. The third column consists of a coma separated list of operations that the user is authorized to perform. Valid values are: • <code>freeze</code>: The user is authorized to to perform DB freeze and thaw operations. • <code>startup_instance</code>: The user is authorized to start a DB instance. • <code>shutdown_instance</code>: The user is authorized to shutdown a DB instance.

There is no reason to add privileged users to this file, as they are automatically authorized.

For more information, refer to the `daemon_user` file.

Controlling Daemon Behavior

The `daemon_options` file contains parameters to control the behavior of the various Solutions Enabler daemons. As each daemon starts, it reads this file and applies all applicable settings.



CAUTION

These parameters are intended for experienced Solutions Enabler users. In most cases, the daemon default settings will be sufficient.

For information on the parameters for each daemon, refer to the daemon's documentation.

The `daemon_options` file is an editable template file located in directory:

<code>/var/symapi/config/</code>	on UNIX systems
<code>\Program Files\EMC\symapi\config\</code>	on Windows systems
<code>SYMAPI\$CONFIG:</code>	on OpenVMS systems

Using a text editor, a System Administrator can add lines to this file using either of the following formats:

<code>NAME = VALUE</code>	Sets the parameter <code>NAME</code> for all daemons that understand this parameter.
<code>stororad:NAME = VALUE</code>	Sets the parameter <code>NAME</code> for only the <code>stororad</code> daemon.
<code>storora*:NAME = VALUE</code>	Sets the parameter <code>NAME</code> for all daemons whose name begins with <code>storora</code> . The <code>*</code> is a wildcard that can be used to match the remainder of a daemon's name.

For more information, refer to the `daemon_options` file.

Base Daemon Support

The base daemon (storapid) provides centralized gatekeeper device management for all Solutions Enabler applications requiring access to Symmetrix arrays, along with the GNS and RDF daemons. This alleviates contention when there are limited gatekeeper resources available and also eliminates the need for every client to constantly select, open, lock, and ping for an available gatekeeper device for every online function.

Additionally, the base daemon monitors Symmetrix External Locks (SEL) and Device External Locks (DEL), and will automatically release any lock(s) held by a crashed application. The base daemon also eliminates the need for Solutions Enabler applications to run as root.

Optionally, you can configure the base daemon to cache inquiry, configuration, and statistical data returned by a Symmetrix array. This increases performance for the calling application by eliminating a request for a gatekeeper from the base daemon's gatekeeper manager, and eliminating a request to the array for data. Data is returned from cache as long as it is considered to be current, as determined by internal caching policies, and externally configurable parameters.

Each host running an instance of the RDF daemon (storrdfd) must also run the base daemon, as it requires the use of the gatekeeper management services. The GNS daemon will use the gatekeeper management services if the base daemon is running, or revert to finding and selecting its own gatekeeper, if not. All other clients and the GNS daemon have the option of setting a system-wide option for using or not using the base daemon services.

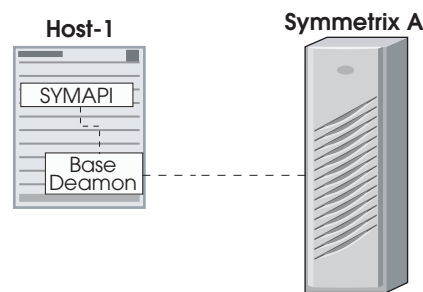


Figure 3-1 Base Daemon

Setting Up the Base Daemon

Use of the base daemon by a client is optional and can be controlled on each host via the daemon options file parameter `client_use`. This parameter has the following possible values:

- ◆ `never` indicates the client is never to connect with the base daemon.
- ◆ `if-running` (default) indicates the client should connect to the base daemon only if it is running.
- ◆ `require` indicates the client is required to always connect to the base daemon. If it is not running, an attempt will be made to start it. If the start fails, all API and CLI functions that communicate to a Symmetrix will return an error until it is started.
- ◆ `optional` indicates the client should connect to the base daemon if running. If it is not running, an attempt is made to start the base daemon. If the start fails, API and CLI functions that communicate to a Symmetrix array will continue to operate.

Starting the Base Daemon

There are three methods for starting the base daemon:

- ◆ Set the daemon to automatically start when the Solutions Enabler libraries first attempt to connect with it by setting the `client_use` daemon option to `OPTIONAL` or `REQUIRE`.

This method can cause a slight delay on the initial connection to the daemon.

- ◆ Manually start the daemon via the `stordaeomon` command line utility as follows:

```
stordaeomon start storapid [-wait Seconds]
```

The `stordaeomon` command requires a path of `/usr/storapi/storbin`.

By default, the `stordaeomon` command waits 30 seconds to verify that the daemon is running. To override this, use the `-wait` option.

- ◆ Set the base daemon to automatically start every time the local host is booted using the following command:

```
stordaeomon install storapid -autostart
```

Pre-starting the daemon, either manually or via the automatic option will eliminate any performance delay incurred when the base daemon needs to be started by an application the first time it tries to connect.

If the base daemon has stopped unexpectedly, it can optionally be restarted automatically by the Solutions Enabler watchdog daemon, storwatchd. A combination of the watchdog mechanism and the auto-start option described above can be used to ensure that the base daemon is always running.

Stopping the Base Daemon

To stop the base daemon, use the following command:

```
stordaeomon shutdown storapid | all [-wait Seconds]
```

Applying the `-all` option will stop all of the daemons currently running.

Base Daemon Optional Behavior Parameters

The daemon options file contains a set of parameters that can be modified to affect base daemon behavior via SYMCLI or SYMAPI commands. The file contains editable behavior parameters set to certain optional defaults in the line entries. Commented lines beginning with a pound sign (#) are ignored.

To remove any parameter option, remove the line entry, rename the file, or comment the line by adding a pound sign (#) at the beginning of the line entry.

Table 3-4 list the possible optional base daemon parameters:

Table 3-4 Base Daemon Optional Behavior Parameters

Optional Behavior Parameter	= <OptValue DefaultValue>	Description
storapid:client_use	never, require, optional if-running	Specifies the client use of the storapid.
storapid:data_caching	disable enable	Enables or disables data caching of Symmetrix configuration and statistical data, as defined by internal caching policies.
storapid:data_compression	disable enable	If set to enable, enables compression/decompression of cached data, reducing memory usage.
storapid:inquiry_timeout	0 - nn, -1 900	Specifies how long inquiry results are to remain in cache before expiring, and new data retrieved from the host and array. A value of -1 indicates the data never expires. A value of zero indicates the data always expires.
storapid:base_timeout	0 - nn, -1 86400	How long Symmetrix data categorized as 'Base' is to remain in cache before expiring, and new data retrieved from the array. 'Base' results are considered those containing configuration type data that rarely, if ever, changes.
storapid:dynamic_timeout	0 - nn, -1 15	How long Symmetrix data categorized as 'Dynamic' is to remain in cache before expiring, and new data retrieved from the array. A value of -1 indicates the data never expires. A value of zero indicates the data always expires. 'Dynamic' results are considered those containing configuration type data subject to frequent or constant change. This value should be adjusted, as necessary, to match the lowest interval option (-i) used with CLI commands.
storapid:statistics_timeout	0 - nn, -1 15	How long Symmetrix data categorized as 'Statistics' is to remain in cache before expiring, and new data retrieved from the array. This value should be adjusted, as necessary, to match the lowest interval option (-i) used with CLI commands. A value of -1 indicates the data never expires. A value of zero indicates the data always expires.

Configuring and Managing the SMI Provider

The Solutions Enabler Storage Management Initiative (SMI) Provider has been paired with the Pegasus CIM Server¹ to provide an SMI-compliant Web-Based Enterprise Management (WBEM) interface for EMC Solutions Enabler products.

If you installed the optional EMC SMI Provider and SMI Libraries, this section explains how to start and stop the Pegasus CIM Server, optionally configure the SSL, and manage the EMC SMI Provider.

For information on supported operating systems, hardware, software, profiles, and subprofiles, refer to the *EMC Solutions Enabler V6.0 Support Matrix*.

Starting and Stopping the Pegasus CIM Server

The Pegasus CIM Server runs in both Windows and UNIX environments. Upon completing the installation, the Pegasus CIM Server is started automatically. You can use the following commands to manually stop and restart the service should the need arise.

Windows

On Windows, the CIM Server runs as a service and can be controlled through the Windows **Services** control panel. The service will not restart at boot unless you modify its startup property in the **Services** control panel to start automatically. The service name is `EMC_SMI_Provider` and it displays as Pegasus CIM Object Manager - `EMC_SMI_Provider` in the **Services** control panel.

The `cimserver.exe` file in the Solutions Enabler `storbin` directory implements the CIM Server. Use the following command to stop the CIM Server:

```
cimserver -stop EMC_SMI_Provider
```

Use the following command to start the CIM Server:

```
cimserver -start EMC_SMI_Provider
```

-
1. The Pegasus CIM Server is developed independently of EMC and therefore is a dependency of the SMI Provider.

UNIX On UNIX, the CIM Server runs as a daemon in the background.

Use the following command from the SYMAPI home directory (./storbin/) to stop the CIM Server:

```
cimserver -s
```

Use the following command from the SYMAPI home directory (./storbin/) to restart the CIM Server:

```
cimserver
```

Controlling the EMC SMI Provider at Runtime

The `EMCProvider.properties` file allows you to configure the behavior parameters of the EMC SMI Provider.

This file is located in the Windows `storbin` directory or the UNIX `storbin` directory.

Table 3-5 EMC SMI Provider Properties

EMC SMI Provider Properties	= <OptValue DefaultValue>	Description
com.emc.storage.provider.database.filename	[filename] <empty>	Specifies the Solution Enabler database filename. If empty, will use the default database file.
com.emc.storage.provider.database.discover	true false	Specifies whether a one time discover is done, upon a cimserver start, prior to processing the first request received by the CIM Server. Note when the CIM Server is started the EMC SMI Provider is not loaded until it is requested from the CIM Server.
com.emc.storage.provider.log.fault.level	0, 1, 3, 2	Specifies the logging level for FAULT and STATUS messages. Possible values are: 0 — FATAL 1 — ERROR 2 — WARNING 3 — INFORMATION If none is specified, the default value is used.

Table 3-5 EMC SMI Provider Properties (continued)

EMC SMI Provider Properties	= <OptValue DefaultValue>	Description
com.emc.storage.provider.log.status.level	0, 1, 3 2	Specifies the logging level for FAULT and STATUS messages. Possible values are: 0 — FATAL 1 — ERROR 2 — WARNING 3 — INFORMATION If none is specified, the default value is used.
com.emc.storage.provider.log.trace.enabled	true false	Specifies whether trace logging should be enabled.
com.emc.storage.provider.log.info	false true	Specifies whether informational logging should be enabled.
com.emc.storage.provider.log.directory	[directory] /var/symapi/log/	Specifies the log directory. If unset or empty, the EMC SMI Provider log file will go to the default location, which is the log directory of the EMC Solutions Enabler. The filename will take the format: emc_provider-yyyymmdd.log

Configuring and Managing CLARiiON Arrays

To fully discover information associated with a CLARiiON array, the username and password for each for the CLARiiON Storage Processor (SP) must be supplied prior to attempting a discovery. This information is provided via the `symcfg authorization` command.

For detailed information about the `symcfg` command, refer to the `symcfg` man page or the *Solutions Enabler Symmetrix Base Management CLI Product Guide*. If you fail to provide valid username and password information for the SP, a discovery will not return metalun, clone, and remote mirror information.

Optional SSL Configuration

Some SAN topologies implement additional security by utilizing SSL verification. You can optionally use your CIM Server via SSL.

If you choose to, you can configure the CIM Server to use an SSL server certificate and private key you have created using a tool such as OpenSSL.

Once you have created the server certificate, place the public certificate in the `server.pem` file and the private key in the `key.pem` file, which are located in the `../symapi/config` Solutions Enabler installation directory. These files can be replaced with new files by simply overwriting the old ones.

For added security, the private key can be stored in a separate location from the certificate file by configuring the CIM Server configuration file, `cimserver_planned.conf`, which is located in `program files\emc\symcli`. For more information on the `cimserver_planned.conf` file, refer to *CIM Server Configuration File* on page 3-43.

The client uses a store of server certificates to identify trusted servers. This store is maintained in the `trusted.crt` file, which is located in the `symapi/config` Solutions Enabler installation directory. After completing the installation, this file only contains the EMC CIM Server certificate. To add additional server certificates, append the `trusted.crt` file with the additional certificate's content using any text editor.

CIM Server Configuration File

On startup, the CIM Server accesses the `cimserver_planned.conf` file to obtain connectivity values. The following properties can be configured:

Table 3-6 CIM Configuration File

Parameter	= <OptValue DefaultValue>	Description
sslCertificateFilePath	<i>[any_valid_path]</i> <code>/var/symapi/config/server.pem^a</code>	Identifies the path to the <code>server.pem</code> file.
sslKeyFilePath	<i>[any_valid_path]</i> <code>/var/symapi/config/key.pem^a</code>	If not set, the private key is expected to be in the same file as the server certificate.
enableHttpsConnection	FALSE TRUE	If set to TRUE, the CIM server will start on the secure port using SSL. If set to FALSE, no secure ports are available.
httpsPort	<i>[valid port number]</i> 5989	Identifies the secure port on which the server should start.
enableHttpConnection	FALSE TRUE	If set to TRUE, the CIM server will start on the non-secure port. If set to FALSE, no non-secure ports are available.
httpPort	<i>[valid port number]</i> 5988	Identifies the non-secure port on which the server should start.
enableAuthentication	TRUE FALSE	Used in conjunction with <code>enableRemotePrivilegedUser</code> parameter. When set to TRUE, a valid CIM username and password is required to perform any operation. CIM users can be created with the <code>cimuser</code> CLI utility as described in <i>Creating a CIM User</i> on page 3-44.
enableRemotePrivilegedUserAccess	TRUE FALSE	Refer to the <code>enableAuthentication</code> parameter.
slp	TRUE FALSE	When set to TRUE, the <code>cimserver</code> will register and activate the SLP Server Agent daemon.

a. The path shown is a UNIX-specific default installation path; your actual install path may differ.

Creating a CIM User

The `cimuser` command is an optional utility to add, modify, remove and list CIM users.

It is only necessary to create CIM Users if you have optionally chosen to use SSL and the `enableAuthentication` and `enableRemotePrivilegedUserAccess` parameters are set to true in the `cimserver_planned_conf` file. For more information on the `cimserver_planned_conf` file, see *CIM Server Configuration File* on page 3-43.

The `cimuser` command enables the management of CIM users. CIM users must be valid users on the local system. If the password is not specified at the command line, the user will be prompted for it.

The following syntax applies:

```
cimuser -a -u username [ -w password ]
cimuser -m -u username [ -w password ] [ -n newpassword]
cimuser -r -u username
cimuser -l
```

This command can be executed by privileged user(s) only.

The `cimuser` command recognizes the following options:

- a
Indicates that a CIM user is to be added.
- m
Indicates that a CIM user's password is to be modified.
- r
Indicates that a CIM user is to be removed.
- l
Displays the names of the CIM users.
- u *username*
Indicates a specific user name.
- w *password*
Indicates the password for the specified user name.
- n *newpassword*
Indicates the new password for the specified user name.

Optional SLP Configuration

The WBEM Discovery using the Service Location Protocol (SLP) specification (DMTF document DSP0205) defines WBEM discovery using the SLP Version 2. SLP is a discovery infrastructure defined by the Internet Engineering Task Force in RFC 2608. The user is expected to have a working knowledge of SLP and WBEM discovery using SLP.

1. Stop the cimserver
2. Set `slp=true` in the `cimserver_planned.conf` file. The default is `false`.
3. Start the cimserver.
4. View the `PegasusStandard.log` file to verify that the SLP registration is an SA. The following is an example of the SLP registration entry in the log file:

```
06/21/04-10:00:19 INFO    cimserver: SLP
Registration Initiated
```

5. Verify that your SLP Client User Agent (which is not provided with Solutions Enabler) can query for a URL and attributes and that the Server Agent responds. Below is an example of a command line client User Agent request and Server Agent response. It does not reflect an exact syntax of requests and responses:

```
slptool findsrvs service:wbem
service:wbem:http://168.159.80.147:5988,65520
service:wbem:https://168.159.80.147:5989,65520

slptool findattrs service:wbem:http://168.159.80.147:5988
(template-url-syntax=http://168.159.80.147:5988),
(service-location-tcp=168.159.80.147),
(service-id=PEG:19745D5D-2F2A-45D0-B6A3-00E01C7B1391),
(service-hi-description=Pegasus CIM Server Version 2.3.2),
(service-hi-name=Pegasus), (template-type=wbem),
(template-version=1.0), (template-description=This template describes
the attributes used for advertising Pegasus CIM Servers.),
(InteropSchemaNamespace=root/emc),
(FunctionalProfilesSupported=Basic Read,Basic Write,Schema
Manipulation,
Instance Manipulation,Association Traversal,Qualifier Declaration,
Indications), (MultipleOperationsSupported=FALSE),

(AuthenticationMechanismsSupported=Basic),
(CommunicationMechanism=cim-xml),
```

```
(ProtocolVersion=1.0),  
(Namespace=root,root/PG_Internal,  
  root/emc,root/PG_InterOp),classinfo,  
(RegisteredProfilesSupported=SNIA:Array,SNIA:Array:Cluster,  
  SNIA:Array:Access Points,SNIA:Array:Disk Drive,SNIA:Array:Location,  
  SNIA:Array:LUN Mapping and Masking,  
  SNIA:Array:Pool Manipulation Capabilities and Settings,  
  SNIA:Array:Extent Mapping,SNIA:Array:LUN Creation,  
  SNIA:Array:Software,SNIA:Server)
```

Uninstalling Solutions Enabler

This chapter explains how to uninstall Solutions Enabler.

- ◆ Overview4-2
- ◆ Uninstalling Solutions Enabler from UNIX4-4
- ◆ Uninstalling Solutions Enabler from Windows4-6
- ◆ Uninstalling Solutions Enabler from OpenVMS4-7
- ◆ Rolling Back an Upgrade4-8

Overview

To uninstall Solutions Enabler, you must first shutdown the application processes that use the Solutions Enabler libraries and binaries, and then uninstall the software.

Stopping the Application Processes

To stop the application processes:

1. Issue the following command to stop the Solutions Enabler daemons:

```
stordaeon shutdown -all
```

For more information on this command, refer to *Stopping Daemons* on page 3-31.

2. Issue the following command to verify that the daemon(s) have stopped:

```
stordaeon list -all
```

For more information on this command, refer to *Viewing Daemons* on page 3-32.

3. Issue the following command to stop the SYMAPI service:

```
symapisrv stop
```

For more information on this command, refer to *Stopping the Server* on page 3-27

4. Issue the following command to verify that the SYMAPI service has stopped:

```
ps -ef | grep symapisrv
```

5. For UNIX, you can also issue the following command to identify any other applications using the Solutions Enabler libraries:

```
fuser /usr/lib/libsym* /usr/lib/libstor*
```

For AIX, issue:

```
fuser -x -f /usr/symcli/shlib/library_name
```

Uninstalling the Software

To uninstall the Solutions Enabler software, refer to the following:

- ◆ For UNIX, refer to *Uninstalling Solutions Enabler from UNIX* on page 4-4.
- ◆ For Windows, refer to *Uninstalling Solutions Enabler from Windows* on page 4-6.
- ◆ For OpenVMS, refer to *Uninstalling Solutions Enabler from OpenVMS* on page 4-7.

Uninstalling Solutions Enabler from UNIX

To uninstall Solutions Enabler from UNIX:



CAUTION

Take care when removing Solutions Enabler, as it may be a prerequisite for other installed products.

1. Verify that all application processes that use the Solutions Enabler libraries and binaries are stopped.

For instructions, refer to *Stopping the Application Processes* on page 4-2.

2. Run the installation program.
3. At the following prompt, enter **u**.

```
What would you like to do? Install (i) Uninstall (u) Register (r) Exit (x): u
```

4. In the HAS BEEN INSTALLED list, determine the item number of the product you want to uninstall:

```
#-----
# The following HAS BEEN INSTALLED in /opt/emc via the emc_install utility.
#-----
# ITEM   PRODUCT                                     VERSION
#-----
01      EMC Solutions Enabler                         V6.0.0
```

and enter it at the following prompt:

```
Enter product number (from HOST) to uninstall: 01
```

5. At the following prompt, enter **y** to confirm your selection or **n** to cancel.

```
The following application will be uninstalled: EMC Solutions Enabler (Run-time) ,
V6.0.0
Is this correct? (y/n): y
```

The install script reminds you that the product is a prerequisite of other installed products, for example:

```
The application to uninstall is required for: EMC ControlCenter
Do you still want to continue? [n]: y
```

6. Enter **y** to uninstall.

The product is removed from the HAS BEEN INSTALLED list.

7. If you are satisfied, enter **x** at the following prompt to exit the script:

```
What would you like to do? Install (i) Uninstall (u) Register (r) Exit (x): x
```

Uninstalling Solutions Enabler from Windows

To uninstall Solutions Enabler from Windows:



CAUTION

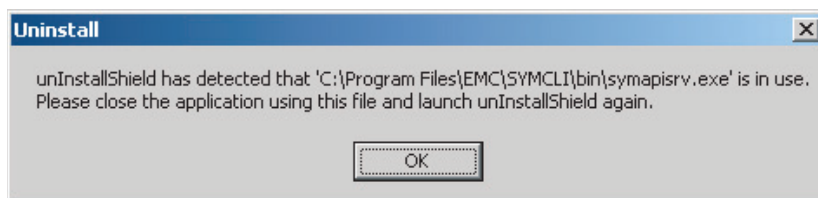
Take care when removing Solutions Enabler, as it may be a prerequisite for other installed products.

1. Verify that all application processes that use the Solutions Enabler libraries and binaries are stopped.

For instructions, refer to *Stopping the Application Processes* on page 4-2.

2. Use the Windows **Add/Remove Programs Properties** dialog box to remove **Solutions Enabler**.

If UnInstallShield detects that a service or library is in use, it will display a message similar to the following. In which case, you should click **OK** to abort the uninstall process, close the application using the service or library, and then restart UnInstallShield.



Uninstalling Solutions Enabler from OpenVMS

To uninstall Solutions Enabler from OpenVMS:



CAUTION

Take care when removing Solutions Enabler, as it may be a prerequisite for other installed products.

1. Verify that all application processes that use the Solutions Enabler libraries and binaries are stopped.

For instructions, refer to *Stopping the Application Processes* on page 4-2.

2. Delete all the files from the installation directory.

Rolling Back an Upgrade

To roll back your upgrade, you must have created copies of the host database and config directories, as explained in *Before You Begin* on page 1-3.

1. Verify that all application processes that use the Solutions Enabler libraries and binaries are stopped.

For instructions, refer to *Stopping the Application Processes* on page 4-2.

2. Export all device groups from the current SYMAPI database:
 - a. Issue a `symdg list` command to list all the device groups.
 - b. Issue a `symdg export` command to export the device groups.

This export is necessary because older versions of Solutions Enabler may not be able to read a database once a newer version of Solutions Enabler has converted it.

For more information on these commands, refer to the *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide*.

3. Install the desired version of Solutions Enabler.

During the installation process, you will be asked to confirm that you want to install an older version than the version currently installed. You must answer **Yes** to this message.

4. Once the installation is complete, issue a `symcfg list` command to verify that the SYMAPI database can be used by the older version:
 - If the database can be used, the rollback is done.
 - If the database cannot be used, issue a `symcfg discover` command to create a Symmetrix host database file, `symapi_db.bin`, and import all the exported device groups.

The rollback is complete.

Host Specific Issues

This section describes the issues in running Solutions Enabler on various hardware platforms. You will find additional information in the Release Notes, which are distributed in hard copy with the Solutions Enabler kits.

The information in this section is organized by hardware platform and operating system.

- ◆ HP-UX Issues A-2
- ◆ Sun Solaris Issues A-6
- ◆ Compaq/Digital UNIX Issues A-7
- ◆ Digital OpenVMS Issues A-9
- ◆ IBM AIX Issues A-13
- ◆ Linux Issues A-15
- ◆ Windows Issues A-16

HP-UX Issues

This section describes the HP-UX system issues concerned with compatibility with the SYMCLI/SYMAPI database file, gatekeeper, and BCV device requirements.

Semaphores for Database and Gatekeeper

The following instructions pertain to examining and/or maximizing semaphore parameters, which control access to the database file and gatekeepers:

To examine or maximizes semaphore parameters for SYMAPI compatibility on HP-UX:

1. Login as `root` and run `SAM`
2. Select **Kernel Configuration, Configurable Parameters**, and examine and/or maximize the following kernel parameters:

maxdsize	1280
sema	1
semmni	128
semmns	256
semmnu	30
semume	10
semaem	16384
semmap	130
semvmx	32767
shmmax	67108864
shmmni	200
shmseg	120

Or select **Actions, Apply Tuned Parameters Set; General OLTP/Database Server System** to maximize a complete set of parameters.

3. Reboot the system to have any modifications take affect.

Creating Pseudo-devices for Gatekeepers and BCVs

If the device you want to use as a gatekeeper or BCV device is accessed through the HP-PB (NIO) SCSI bus controller and you want the device to be visible to your host, you must create a pseudo-device for that device. (A pseudo-device is necessary for every device you want visible to the host.)

Your HP-UX operating system may require a patch to support the HP-PB (NIO) SCSI board. Patches for the HP-PB SCSI Pass-Thru driver (spt0) are available for HP-UX V11.0 and higher from HP on an Extension Media CD. Consult your HP representative about spt drivers for your specific system.

If your HP system is configured with an HSC fast-wide differential SCSI interface board and a device accessed through the HSC SCSI bus is available, you can specify the gatekeeper devices through the procedure outlined in the *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide*.

To create pseudo-devices and specify devices as gatekeepers and BCV devices, do the following:

1. Execute the `ioscan` command and find the full pathnames of the gatekeeper and BCV devices.

For example, the full pathname of the Symmetrix volume designated to be the gatekeeper is `/dev/rdisk/clt2d1`.

2. Enter the `lsdev` command and note the output. For example:

```
lsdev -d spt0
Character      Block      Driver      Class
       75          -1       spt0        spt
```

The wide SCSI Pass-Thru driver is identified as **spt0**. If there is no output in response to this command, the **spt0** driver is missing. Install the proper driver before proceeding.

Note that there is also an spt driver. The spt driver will not work in this environment.

3. Create the device node for the gatekeeper device.

This step creates a pseudo-device that is incapable of functioning like a normal device. It can only be used as a gatekeeper device or to process TimeFinder control functions directed to a BCV device.

For example, to create the device node:

```
mknod /dev/rdisk/pseudo_c1t2d1 c 75 0x012100
```

Where:

`/dev/rdisk/pseudo_c1t2d1` is the full pathname of the pseudo-device associated with `/dev/rdisk/c1t2d1`.

`c` specifies character (raw) device node creation

`75` is the character value from the output of the `lsdev` command. This is the major number of the device file.

`0x012100` is the minor number of the device file. The individual values of the minor number are:

`0x` indicates that the number is hexadecimal.

`01` is the hexadecimal number of the controller referenced by `/dev/rdisk/c1t2d1`

`2` is the hexadecimal number of the target ID referenced by `/dev/rdisk/c1t2d1`

`1` is the hexadecimal number of the LUN referenced by `/dev/rdisk/c1t2d1`

`00` must be the last two digits of the minor number.

4. Repeat step 3 for all BCV devices and alternate gatekeeper devices.



CAUTION

Do not perform I/O through the device (`/dev/rdisk/cxtxdx`) associated with the pseudo-device, nor use the pseudo-device as a normal device. If you do, you have two paths to the same device from two different device drivers. Unknown results may occur.

5. To create the mapping information of standard devices to pseudo-devices, create the file:

```
/var/symapi/config/pseudo_devices
```

For each gatekeeper and BCV device, add a mapping to a pseudo-device. For example, in the `pseudo_devices` file, add the following line to map the pseudo-device filename (in **bold**), to the Symmetrix device file.

```
/dev/rdsk/c1t0d0          /dev/rdsk/pseudo_c1t0d0
```

SYMAPI will then use this pseudo-device instead of the physical device file name.

When the `SymDiscover()` function is used, the pseudo-device mappings get posted in the log file

(`/var/symapi/log/symapi*.log`).

Sun Solaris Issues

The following instructions pertain to examining and/or maximizing semaphore parameters, which control access to the database file and gatekeepers:

To examine or modify semaphore parameters for SYMCLI/SYMAPI compatibility on Solaris, reference or add semaphore parameters at the end of `/etc/system`.

For example:

```
set semsys:seminfo_semmni=600
set semsys:seminfo_semmns=600
set semsys:seminfo_semume=600
set semsys:seminfo_semmnu=600
set shmsys:shminfo_shmmax=524288
set shmsys:shminfo_shmmni=600
```

For Solaris 8 and above, set `shmsys:shminfo_shmmax=1048576`

If parameter values have been modified, reboot the system to have them take affect.

Compaq/Digital UNIX Issues

The following instructions pertain to examining and/or maximizing semaphore parameters, which control access to the database file and gatekeepers.

- ◆ For versions 5.0 and later, use the provided system utility `dxkerneltuner` to modify these values.
- ◆ For versions prior to version 5.0, reference or add semaphore parameters at the end of path `/etc/sysconfigtab`.

For example:

```
ipc:
shm-max=2118123520
shm-seg=32
sem-aem=16384
sem-mni=200
num-of-sems=200
sem-msl=200
sem-opm=200
sem-ume=200
sem-vmx=32767
```

If parameter values have been modified, reboot the system to have them take affect.

Compaq TruCluster Server

For Compaq Tru64 UNIX Clusters (utilizing clustered filesystems), the executable images can be shared but the Solutions Enabler database, log, and config files must be unique for each node in the cluster. In this environment, each node in the cluster has local needs.

An example post-install script provided in this section should be run as a post installation step of the EMC Solutions Enabler software. It will determine if the node executing the script is part of a Cluster. The system administrator should take the defaults, during the installation, and then execute this script afterward or just execute the command interactively if they know this is a Tru64 Cluster environment.

For each node, the script creates a private area in the cluster for the files that are unique for Solutions Enabler.

Use the following Compaq command (`mkcdsl`) to invoke the script:

```
/usr/sbin/mkcdsl -a -f /var/symapi
```

where:

`-a`
Specifies all members.

`-f`
Forces the overwriting of the existing CDSL or member-specific file or directory. When the force option is used with a copy option, `mkcdsl` will overwrite an existing member-specific file or directory. Without the force (`-f`) option, `mkcdsl` issues an error or message whenever the physical path of the target differs from the specified targetname (for example, when targetname resolution traverses a symbolic link), or when the source for a specified copy option cannot be found. Unless the `-f` option is specified, `mkcdsl` will exit when it encounters a situation that would generate an error message. The `mkcdsl` command issues a warning message if the specified sourcename differs from the calculated sourcename. However, you do not need the `-f` option to stop `mkcdsl` from exiting when it encounters a situation that generates a warning message.

Cluster Script Example

```
#!/usr/bin/ksh

/sbin/sysconfig -q generic clu_active_member >/dev/null
2>&1

cluster_member=$?

if [[ $cluster_member -eq 0 ]]
then
    /usr/sbin/mkcdsl -f -a -c /var/symapi
fi
```

Digital OpenVMS Issues

The following instructions pertain to examining SYMCLI return codes for the OpenVMS operating system.

A set of return codes for the various conditions possible with each SYMCLI/CLARCLI command are provided for UNIX and Windows platforms. However, for the OpenVMS platforms, discernible return codes are not yet available and, therefore, require interpolation or special processing of the returned hexadecimal value (resulting from a \$STATUS query). This is because the format of the OpenVMS return value also includes a severity level field in the three least significant bits (00-02). The return code is described in the next field (03-15). Table A-1 describes the set of possible return hex values and their associated meaning with SYMCLI/CLARCLI on OpenVMS.

Table A-1 OpenVMS Return Code Interpolation Table

Returned Hex Value ^{1aa}	SYMCLI Return Code	OpenVMS Severity Level	SYMCLI Name
%X1FFF0001	00	1 (S)	SUCCESS
%X1FFF000C	01	4 (F)	FAIL
%X1FFF0012	02	2 (E)	DB_FILE_IS LOCKED
%X1FFF001A	03	2 (E)	SYM_IS_LOCKED
%X1FFF0023	04	3 (I)	NOT_ALL_SYNCHRONIZED
%X1FFF002B	05	3 (I)	NONE_SYNCHRONIZED
%X1FFF0033	06	3 (I)	NOT_ALL_UPDATED
%X1FFF003B	07	3 (I)	NONE_UPDATED
%X1FFF0043	08	3 (I)	NOT_ALL_PINGED
%X1FFF004B	09	3 (I)	NONE_PINGED
%X1FFF0053	10	3 (I)	NOT_ALL_SYNCHED
%X1FFF005B	11	3 (I)	NONE_SYNCHED
%X1FFF0063	12	3 (I)	NOT_ALL_RESTORED
%X1FFF006B	13	3 (I)	NONE_RESTORED
%X1FFF0073	14	3 (I)	NOT_ALL_VALID

Table A-1 OpenVMS Return Code Interpolation Table (continued)

Returned Hex Value ^{1aa}	SYMCLI Return Code	OpenVMS Severity Level	SYMCLI Name
%X1FFF007B	15	3 (I)	NONE_VALID
%X1FFF0083	16	3 (I)	SYM_NOT_ALL_LOCKED
%X1FFF008B	17	3 (I)	SYM_NONE_LOCKED
%X1FFF0093	18	3 (I)	ALREADY_IN_STATE
%X1FFF009A	19	2 (E)	GK_IS_LOCKED
%X1FFF00A2	20	2 (E)	WP_TRACKS_IN_CACHE
%X1FFF00AA	21	2 (E)	NEED_MERGE_TO_RESUME
%X1FFF00B2	22	2 (E)	NEED_FORCE_TO_PROCEED
%X1FFF00BA	23	2 (E)	NEED_SYMFORCE_TO_PROCEED
%X1FFF00C3	24	3 (I)	NOT_IN_SYNC
%X1FFF00CB	25	3 (I)	NOT_ALL_SPLIT
%X1FFF00D3	26	3 (I)	NONE_SPLIT
%X1FFF00DB	27	3 (I)	NOT_ALL_SYNCINPROG
%X1FFF00E3	28	3 (I)	NONE_SYNCINPROG
%X1FFF00EB	29	3 (I)	NOT_ALL_RESTINPROG
%X1FFF00F3	30	3 (I)	NONE_RESTINPROG
%X1FFF00FB	31	3 (I)	NOT_ALL_SUSPENDED
%X1FFF0103	32	3 (I)	NONE_SUSPENDED
%X1FFF010B	33	3 (I)	NOT_ALL_FAILED_OVER
%X1FFF0113	34	3 (I)	NONE_FAILED_OVER
%X1FFF011B	35	3 (I)	NOT_ALL_UPDATEINPROG
%X1FFF0123	36	3 (I)	NONE_UPDATEINPROG
%X1FFF012B	37	3 (I)	NOT_ALL_PARTITIONED
%X1FFF0133	38	3 (I)	NONE_PARTITIONED
%X1FFF013B	39	3 (I)	NOT_ALL_ENABLED

Table A-1 OpenVMS Return Code Interpolation Table (continued)

Returned Hex Value ^{1aa}	SYMCLI Return Code	OpenVMS Severity Level	SYMCLI Name
%X1FFF0143	40	3 (I)	NONE_ENABLED
%X1FFF014B	41	3 (I)	NOT_ALL_SYNCHRONIZED_AND_ENABLED
%X1FFF0153	42	3 (I)	NONE_SYNCHRONIZED_AND_ENABLED
%X1FFF015B	43	3 (I)	NOT_ALL_SUSP_AND_ENABLED
%X1FFF0163	44	3 (I)	NONE_SUSP_AND_ENABLED
%X1FFF016B	45	3 (I)	NOT_ALL_SUSP_AND_OFFLINE
%X1FFF0173	46	3 (I)	NONE_SUSP_AND_OFFLINE
%X1FFF017A	47	2 (E)	WONT_REVERSE_SPLIT
%X1FFF0182	48	2 (E)	CONFIG_LOCKED
%X1FFF018A	49	2 (E)	DEVS_ARE_LOCKED
%X1FFF0192	50	2 (E)	CLI_C_MUST_SPLIT_PROTECT
%X1FFF019A	51	2 (E)	CLI_C_PAIRIED_WITH_A_DRV
%X1FFF01A2	52	2 (E)	CLI_C_PAIRIED_WITH_A_SPARE
%X1FFF01AB	53	3 (I)	CLI_C_NOT_ALL_COPYINPROG
%X1FFF01B3	54	3 (I)	CLI_C_NONE_COPYINPROG
%X1FFF01BB	55	3 (I)	CLI_C_NOT_ALL_COPIED
%X1FFF01C3	56	3 (I)	CLI_C_NONE_COPIED
%X1FFF01CB	57	3 (I)	CLI_C_NOT_ALL_COPYONACCESS
%X1FFF01D3	58	3 (I)	CLI_C_NONE_COPYONACCESS
%X1FFF01DA	59	2 (E)	CLI_C_CANT_RESTORE_PROTECT
%X1FFF01E3	60	3 (I)	CLI_C_NOT_ALL_CREATED
%X1FFF01EB	61	3 (I)	CLI_C_NONE_CREATED
%X1FFF01F3	62	3 (I)	CLI_C_NOT_ALL_CREATED
%X1FFF01FB	63	3 (I)	CLI_C_NONE_CREATED
%X1FFF0202	64	2 (E)	CLI_C_STD_BKGRND_SPLIT_IN_PROG

Table A-1 OpenVMS Return Code Interpolation Table (continued)

Returned Hex Value ^{1aa}	SYMCLI Return Code	OpenVMS Severity Level	SYMCLI Name
%X1FFF020A	65	2 (E)	CLI_C_SPLIT_IN_PROG
%X1FFF0213	66	3 (I)	CLI_C_NOT_ALL_COPYONWRITE
%X1FFF021B	67	3 (I)	CLI_C_NONE_COPYONWRITE

- a. For OpenVMS, use `write sys$output $status` to view return code. The following special DCL program can automatically convert these OpenVMS values to the proper SYMCLI/CLARCLI return codes:

[SAMPLE-DCL]

```
$ ! Example: Convert SYMCLI return codes.
$ !
$ a = ( %x0000ffff .and. 'p1) ! Mask off bits 16-31.
$ a = ( a/8 ) ! Shift 3-15 right.
$ write sys$output 'a ! Print return code without
$ ! severity level.
$ !
```

For example, an OpenVMS %X1FFF002B value converts to a 5 return code.

IBM AIX Issues

On IBM AIX systems for SYMCLI/SYMAPI database and gatekeeper access, the upper parameter limits are preset to 4096 for semaphore and shared memory mechanisms (IDs). These are controlled by data structures, which are allocated and deallocated on demand by the kernel. Therefore, there are no optional controls to modify these limits.

Memory Access

The two principal limits of concern are set to 4096 for the maximum number of semaphore and shared memory IDs.

Oracle Database Mapping

Oracle 8 database mapping with SYMCLI is supported on 32-bit AIX V4.3 and above.

You may need to create the Oracle library, `libclntsh.so`.

To determine if the library exists **for Oracle 8**, execute the following:

```
ls $ORACLE_HOME/lib/libclntsh.so
```

If the library does not exist, execute the following make command:

```
make -f $ORACLE_HOME/rdbms/lib/ins_rdbms.mk  
client_sharedlib
```

The Oracle 8 OCI executable is linked dynamically. You must set the following environment variable as follows:

```
setenv LIBPATH $ORACLE_HOME/lib
```

BCV Devices Lost after Reboot

When a system comes back up after a reboot, it will not recognize your mapped BCVs. To work around this problem, you should run the following special BCV script (mkbcv):

```
cd /
./inq.AIX | more (look for no gaps in the numbers, ie..
                 rhdisk0, rhdisk1, rhdisk3... - rhdisk2 is missing)
cd /usr/lpp/Symmetrix/bin
./mkbcv -a ALL
cd /
./inq.AIX | more (look for no gaps in the numbers, ie..
                 rhdisk0, rhdisk1, rhdisk2... - rhdisk2 is not
                 missing)
```

It is recommended to have `./mkbcv -a ALL` in your AIX boot procedures.

inq.AIX can be found on the EMC FTP site.

Linux Issues

This section addresses the Linux issues.

Mounting a CD Over a Network

To install Solutions Enabler in a Linux on S390 environment, you must mount the CD over a network. To do this, place the CD in the CD-ROM drive of another host, and make it available for sharing.

For example:

1. From a Solaris host, enter the following in `/etc/dfstab`:

```
share -F nfs -o ro -d SYMCLI/cdrom/cdrom0
```

2. Issue the command:

```
shareall
```

3. From the Linux for S390 host, enter the following command to mount the remote CD:

```
mount -t nfs -o ro hostname:/cdrom/cdrom0/mnt
```

The CD is now mounted. Continue the installation with *Step 2: Run the Install Script* on page 2-3, specifying `/mnt` for the *<CD-ROM mount point>*.

Installing the Kernal Patch

In a Linux S390 installation, you must download and install a kernel patch. For instructions, refer to *Install the Kernel Patch* on page 2-10.

Windows Issues

For Windows platforms in a clustered environment, the disk drive (device) in the Symmetrix array that you assign as gatekeeper must be a minimum of 8 MB in size and have a signature.

In a non-clustered environment, gatekeeper devices smaller than 8 MB will show up in the new Disk Manager as devices with no available information. (Disk Manager just displays the Disk number and a blank bar.) The devices are still addressable at the SCSI level, and SYMCLI scripts continue to work. (There may be some implications for device naming, since the Windows Device Manager does not create some of the normal device objects for devices smaller than 8 MB.

Creating Daemon Auto-Start Scripts

This appendix explains how to set daemons to automatically start when a system is rebooted.

The procedures in this appendix use the stororad daemon as an example. To use these procedures with other Solutions Enabler daemons, simply replace `stororad` with the name of another daemon.

For more information on Solutions Enabler daemons, refer to *Setting Up Daemons for Distributed Application Support* on page 3-30.

The procedures in this appendix are organized by operating system.

- ◆ Creating an Auto-Start Script for Solaris B-2
- ◆ Creating an Auto-Start Script for Linux B-3
- ◆ Creating an Auto-Start Script for HP-UX B-4
- ◆ Creating an Auto-Start Script for IBM AIX B-6
- ◆ Creating an Auto-Start Script for Windows B-8

Creating an Auto-Start Script for Solaris

To create an auto-start script for Solaris, do the following:

1. Create a file named `stororad` and add the following to it:

```
#!/bin/sh

#Where the stordaeomon utility resides.
BINDER=/usr/storapi/storbin

case $1 in
start)
    echo "Starting stororad"
    $BINDER/stordaeomon start stororad
    ;;

stop)
    echo "Stopping stororad"
    $BINDER/stordaeomon shutdown stororad -wait 10
    ;;

*)
    echo "[stororad start script] Usage: $0 {start|stop}"
    exit 1
    ;;

esac
exit 0
```

2. Copy the file to `/etc/init.d` and make sure that it is owned by root and that root has read/execute access to it by entering the following as root:

```
cp stororad /etc/init.d
chmod a+x /etc/init.d/stororad
```

3. Create links to the file from `/etc/rc.d/rc3.d` by entering the following:

```
ln /etc/init.d/stororad /etc/rc3.d/S999stororad

ln /etc/init.d/stororad /etc/rc3.d/K999stororad
```

To create symbolic links instead of hard links, use the `ln -s` option.

Use any runlevel (3 and above) appropriate for your environment.

The priority lever (999 and above) is not critical and can be replaced with a different value.

Creating an Auto-Start Script for Linux

To create an auto-start script for Linux, do the following:

1. Create a file named `stororad` and add the following to it:

```
#!/bin/sh

#Where the stordaeomon utility resides.
BINDER=/usr/storapi/storbin

case $1 in
start)
    echo "Starting stororad"
    $BINDER/stordaeomon start stororad
    ;;

stop)
    echo "Stopping stororad"
    $BINDER/stordaeomon shutdown stororad -wait 10
    ;;

*)
    echo "[stororad start script] Usage: $0 {start|stop}"
    exit 1
    ;;

esac
exit 0
```

2. Copy the file to `/etc/rc.d/init.d` and make sure that it is owned by root and that root has read/execute access to it by entering the following as root:

```
cp stororad /etc/rc.d/init.d
chmod a+x /etc/rc.d/init.d/stororad
```

3. Create links to the file from `/etc/rc.d/rc3.d` by entering the following:

```
ln -s/etc/rc.d/init.d/stororad /etc/rc.d/rc3.d/
S99stororad

ln -s/etc/rc.d/init.d/stororad /etc/rc.d/rc3.d/
K99stororad
```

Use any runlevel (3 and above) appropriate for your environment.

The priority lever (99 and above) is not critical and can be replaced with a different value.

Creating an Auto-Start Script for HP-UX

To create an auto-start script for HP-UX, do the following:

1. Create a file named `stororad` and add the following to it:

```
#!/bin/sh

#Where the stordaeomon utility resides.
BINDER=/usr/storapi/storbin

case $1 in
start)
    echo "Starting stororad"
    $BINDER/stordaeomon start stororad
    ;;

stop)
    echo "Stopping stororad"
    $BINDER/stordaeomon shutdown stororad -wait 10
    ;;

start_msg)
    echo "Start the SYMCLI stororad daemon"
    ;;

stop_msg)
    echo "Stop the SYMCLI stororad daemon"
    ;;

*)
    echo "[stororad start script] Usage: $0 {start|stop}"
    exit 1
    ;;

esac
exit 0
```

2. Copy the file to `/sbin/init.d` and make sure that it is owned by root and that root has read/execute access to it by entering the following as root:

```
cp stororad /sbin/init.d
chmod a+x /sbin/init.d/stororad
```

3. Create links to the file from `/sbin/rc3.d` by entering the following:

```
ln -s/sbin/init.d/stororad /sbin/rc3.d/S999stororad
```

```
ln -s/sbin/init.d/stororad /sbin/rc3.d/K999stororad
```

Use any runlevel (3 and above) appropriate for your environment.

The priority lever (999 and above) is not critical and can be replaced with a different value.

Creating an Auto-Start Script for IBM AIX

The procedure for creating an auto-start script for IBM AIX depends on whether you have the *run level scripting* feature installed.

Run Level Scripting — Installed

If you have the *run level scripting* feature installed, do the following to create the auto-start script:

1. Create a file named `stororad` and add the following to it:

```
#!/bin/ksh

#Where the stordaeon utility resides.
BINDER=/usr/storapi/storbin

case $1 in
start*)
    echo "Starting stororad"
    $BINDER/stordaeon start stororad
    ;;

stop)
    echo "Stopping stororad"
    $BINDER/stordaeon shutdown stororad -wait 10
    ;;

*)
    echo "[stororad start script] Usage: $0 {start|stop}"
    exit 1
    ;;
esac
exit 0
```

2. Copy the file to `/etc/init.d` and make sure that it is owned by root and that root has read/execute access to it by entering the following as root:

```
cp stororad /etc/init.d
chmod a+x /etc/init.d/stororad
```

3. Create links to the file from `/sbin/rc2.d` by entering the following:

```
ln -s/etc/init.d/stororad /etc/rc2.d/S999stororad
```

```
ln -s/etc/init.d/stororad /sbin/rc2.d/K999stororad
```

Use any runlevel (2 and above) appropriate for your environment.

The priority level (999 and above) is not critical and can be replaced with a different value.

Run Level Scripting — Not Installed

If you do NOT have the *run level scripting* feature installed, insert an entry into the `/etc/inittab` file:

```
mkitab "stororad:2:once:/usr/storapi/bin/stordaeomon  
start stororad"
```

If a `stororad` entry already exists in the table, you can use `rmitab stororad` to remove it.

Use any runlevel (2 and above) appropriate for your environment. If appropriate, you can specify multiple runlevels.

Creating an Auto-Start Script for Windows

In Windows, Solutions Enabler daemons run as services.

To create an auto-start script for Windows, do the following:

1. Start the daemon at least once to ensure that it is registered as a service:

```
stordaeomon start stororad
```

By default, services are configured with a *Manual* startup mode.

2. Invoke the Service Manager by selecting **Control Panel, Administrative Tools, Services**.
3. Double-click the **stororad** service and change the startup type from *Manual* to *Automatic*.
4. Click **OK** to save the changes.

Solutions Enabler Directories

This appendix contains the directory list for UNIX and Windows installations.

- ◆ UNIX Directories..... C-2
- ◆ Windows Directories C-3

UNIX Directories

Table C-1 lists the directories for UNIX platforms. Your directories may differ from this list since the location of these directories is configurable at installation.

Table C-1 UNIX Directories

Contents	Directories	Details
Binaries for executables	/usr/storapi/storbin /usr/storapi/bin	STORCLI binaries SYMCLI binaries
Shared libraries	/usr/storapi/shlib	All shared libraries
Database engines	/usr/storapi/shlib/sql/IBMUDB/ /usr/storapi/shlib/sql/INFORMIX/ /usr/storapi/shlib/sql/ORACLE/ /usr/storapi/shlib/sql/SYBASE/	IBM database engine Informix database engine Oracle database engine Sybase database engine
Language interfaces	/usr/storapi/interfaces/java/ /usr/storapi/interfaces/xml/c_xml/examples/ /usr/storapi/interfaces/xml/c_xml/include /usr/storapi/interfaces/xml/shlib/ /usr/storapi/interfaces/xml/java_xml/ /usr/storapi/interfaces/xml/java_xml/examples/ /usr/storapi/interfaces/xml/schemas /usr/storapi/interfaces/xml/jni/	Java language interface XML examples for C programmers XML API header files XML shared libraries Jar files for XML XML examples for Java programmers XML schema files JNI examples and Java docs
SYMCLI manpages	/usr/storapi/storman/man1 /usr/storapi/storman/man3 /usr/storapi/man/man1 /usr/storapi/man/man3	STORCLI manpages STORAPI manpages SYMCLI manpages SYMAPI and CLARAPI manpages
Daemons	/usr/storapi/deamons/	Location of the database daemons
Configuration database file(s)	/var/symapi/db/	Contains the configuration database file(s) for SYMAPI, CLARAPI, and STORAPI
SYMAPI environment and system files	/var/symapi/config	Includes licenses, avoidance, options, and server network files
XML Properties	/var/symapi/interfaces/xml/java_xml/	XML properties files for SYMAPI and STORAPI
SYMAPI log files	/var/symapi/log	Contains log of significant events

Windows Directories

Table C-2 lists the default directories for Windows. Your directories may differ from this list since the location of these directories is configurable at installation.

Table C-2 Windows Directories

Contents	Directory	Details
Binaries for executables	c:\Program Files\EMC\SYMCLI\storbin c:\Program Files\EMC\SYMCLI\bin	STORCLI binaries SYMCLI binaries
Shared libraries	c:\Program Files\EMC\SYMCLI\shlib	All shared libraries
Database engines	c:\Program Files\EMC\SYMCLI\shlib\sql\IBMUDB c:\Program Files\EMC\SYMCLI\shlib\sql\Oracle c:\Program Files\EMC\SYMCLI\shlib\sql\SQLSERVER	IBM database engine Oracle database engine SQL server database engine
Language interfaces	c:\Program Files\EMC\SYMCLI\interfaces\java c:\Program Files\EMC\SYMCLI\interfaces\xml\c_xml\examples c:\Program Files\EMC\SYMCLI\interfaces\xml\c_xml\include c:\Program Files\EMC\SYMCLI\interfaces\c_xml\shlib c:\Program Files\EMC\SYMCLI\interfaces\dotnet_xml\examples c:\Program Files\EMC\SYMCLI\interfaces\java_xml\examples c:\Program Files\EMC\SYMCLI\interfaces\schemas c:\Program Files\EMC\SYMCLI\interfaces\jni	Java language interface XML examples for C programmers XML API header files XML shared libraries Jar files for XML XML examples for Java programmers XML schema files JNI examples and Java docs
SYMCLI manpages	c:\Program Files\EMC\SYMCLI\storman\man1 c:\Program Files\EMC\SYMCLI\storman\man3 c:\Program Files\EMC\SYMCLI\man\man1 c:\Program Files\EMC\SYMCLI\man\man3	STORCLI manpages STORAPI manpages SYMCLI manpages SYMAPI and CLARAPI manpages
Daemons	c:\Program Files\EMC\SYMCLI\daemons	Location of the database daemons
Connectivity directories	c:\Program Files\EMC\SYMCLI\conn c:\Program Files\EMC\SYMCLI\conn\bin c:\Program Files\EMC\SYMCLI\conn\data c:\Program Files\EMC\SYMCLI\conn\etc c:\Program Files\EMC\SYMCLI\conn\nt_lib	Connectivity related binaries, data, and database libraries
Configuration database file(s)	c:\Program Files\EMC\SYMAPI\db	Contains the configuration database file(s) for SYMAPI, CLARAPI, and STORAPI

Table C-2 Windows Directories (continued)

Contents	Directory	Details
SYMAPI environment and system files	c:\Program Files\EMC\SYMAPI\config	Includes licenses, avoidance, options, and server network files
XML Properties	c:\Program Files\EMC\SYMAPI\interfaces\xml\java_xml	XML properties files for SYMAPI and STORAPI
SYMAPI log files	c:\Program Files\EMC\SYMAPI\log	Contains log of significant events
Providers	c:\Program Files\EMC\SYMCLI\shlib	VSS and VDS providers

A

- autorun, CD-ROM 2-12
- avoidance files 3-14

C

- CD-ROM
 - mounting in UNIX 2-2
 - unmounting from UNIX 2-10
 - Windows autorun 2-12
- certificate files, security 3-22
- CIM Server
 - configuration file 3-43
 - installation option 2-15
 - starting and stopping 3-39
- CIM user, creating 3-44
- CLARiiON, discovery setup 3-41
- CLI path, setting 3-6
- client installs 1-4
- client/server security 1-5
 - configuring the client 3-20
 - configuring the server 3-21
 - security levels 3-19
- Compaq issues A-7
- components, enabling functionality 3-2

D

- daemon_options file 3-34
- daemon_users file 3-32
- daemons 3-30
 - base daemon support 3-35
 - configuring to auto-start on boot 3-32
 - controlling 3-34

- setting to auto-start at boot B-1
 - starting 3-31
 - stopping 3-31
 - viewing 3-32
- database file A-2
- database locks 3-11
- Device External Locks (DEL) 3-35
- Digital UNIX issues A-7
- Distributed Lock Manager, OpenVMS 3-13

E

- environment variables, setting 3-5

F

- files
 - avoidance 3-14
 - license 3-4
 - netcnfg 3-23
 - nethost 3-24
 - network services 3-19
 - options 3-16
 - selection 3-14

G

- gatekeeper devices 3-8
 - choosing 3-8
 - dedicated 3-10
 - identifying 3-8
 - listing 3-9
 - locking 3-11
 - sizing 3-10
- gkavoid file 3-14

gkselect file 3-14

H

help path, setting 3-7

HP-UX issues A-2

I

inqfile file 3-15

installation

help files 3-7

man pages 3-7

mutexes for Windows NT 3-13

OpenVMS 2-20

rolling back 4-8

semaphore requirements for UNIX 3-11

UNIX 2-2

UNIX directories 2-3

Windows 2-12

instance identifier 3-32

J

Java interface component 2-8

K

kernel, S390 patch 2-10

L

license file 3-4

license keys 3-2

License Management Facility (LMF), invoking 3-3

Linux

issues A-15

starting the SCSI generic driver 3-5

Linux on S390

installing the kernel patch 2-10

mounting the CD over a network A-15

locking

OpenVMS 3-13

Windows 3-13

M

man pages 3-7

monitoring option, installing on Windows 2-17

N

netcnfg file

editing 3-23

reading 3-26

setting security level 3-20

nethost file, configuring 3-24

network services 3-19

O

OpenVMS

installing in 2-20

issues A-9

locking 3-13

uninstalling from 4-7

optional behavior parameters 3-16

optional libraries

installing in UNIX 2-6

installing in Windows 2-15

options file 3-16

setting security level 3-20

options, removing defaults 3-18

Oracle multiple instances 3-28

Oracle on AIX issues A-13

Oracle remote server 3-28

P

parameters, optional behavior 3-16

permissions, setting 3-5

private key, SSL 3-42

pseudo-devices, creating A-3

public key, SSL 3-42

R

RDBMS environment variables 3-29

requirements 1-3

S

s390ioctl.o module, inserting into the kernel 2-10

SCSI generic driver, starting 3-5

Secure Socket Layer (SSL) 1-5

security, client/server 1-5

compatibility between versions 3-22

- configuring the client 3-20
 - configuring the server 3-21
 - security levels 3-19
- semaphore identifier 3-12
- semaphores 3-11, A-2
 - de-allocating 3-12
 - refreshing 3-12
 - requirements 3-12
- server installs 1-4
- Service Location Protocol (SLP) 3-45
- SLP, configuraiton 3-45
- SMI Provider 3-39
 - installation option 2-15
 - parameters, runtime 3-40
- SPEA Root certificate file 3-22
- SSL 3-42
 - certificate usage 3-42
 - configuring 3-42
 - private key 3-42
- storapid daemon
 - building a non_root user id file 3-32
 - optional parameters 3-37
 - setting up 3-36
 - starting 3-36
 - stopping 3-37
 - support 3-35
- Sun Solaris issues A-6
- SYMAPI database, building 3-5
- SYMAPI server 3-19
 - certificate file 3-22
 - installing 1-4
 - registering in the Windows service manager 3-24
 - starting 3-25
 - stopping 3-27
 - uninstalling 3-27
- symapisrv_cert.pem file 3-22
- symavoid file 3-15
- symcfg discover command 3-5

- error 2-10
- SYMCLI 1-2
- symgate command 3-9
- syminq command 3-8
- Symmetrix External Locks (SEL) 3-35
- Symmetrix Integration Utility (SIU) service,
 - installing 2-18
- sympd command 3-10

T

- temporary files, removing 2-10
- trusted-user host access file (nethost), configuring 3-24

U

- uninstall 4-2
- UNIX
 - CD-ROM mount commands 2-2
 - installation directories 2-3
 - installing in 2-2
 - mount point 2-3
 - uninstalling from 4-4
- upgrade
 - OpenVMS 2-20
 - rolling back 4-8
 - UNIX 2-2
 - Windows 2-12

W

- WBEM Discovery 3-45
- Windows
 - installing in 2-12
 - issues A-16
 - locking 3-13
 - uninstalling from 4-6
- write access, setting 3-5

