CORBA Gateway Administration Guide

Solstice Enterprise Manager™ 4.1

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303
U.S.A. 650-960-1300

Part No. 806-7976-10
October 2001, Revision A

Please
Recycle

Adobe PostScript

Contents

Figures

Tables

Preface

The *CORBA Gateway Administration Guide* provides information on installing, configuring and using the SEM CORBA ToolKit to build and package the SEM CORBA Gateways. It contains a introduction to CORBA Gateways, followed by an overview of the ToolKit, and the subsequent chapters describe the administration tasks and how to carry out these tasks on each of the SEM CORBA Gateway components.

Who Should Use This Book

This document is intended for SEM CORBA system administrators and application development engineers. Application development engineers may refer to this guide to build and package SEM CORBA Gateway executables and find out how to fine tune configuration parameters and use debug options.

Before You Read This Book

This book assumes:

- A good understanding of Solstice EM (including some experience)
- A basic knowledge of CORBA

How This Book Is Organized

This book contains the following chapters:

Chapter 1 "**Introduction**" explains what a CORBA Gateway is, and lists the tasks involved in its administration.

Chapter 2 "**SEM CORBA ToolKit**" explains how to compile, build and package the SEM CORBA Gateway using the ToolKit.

Chapter 3 "**Administering the System**" explains how to configure, start, stop and troubleshoot Gateway components.

Chapter 4 "**Improving CORBA Gateway Performance**" explains how to optimize and improve the performance of CORBA Gateways.

Related Books

Refer to the following publications for additional information about related topics:

- *Management Information Server (MIS) Guide*
- *Managing Your Network*
- *Customizing Guide*
- *Developing CORBA Applications*

What Typographic Changes Mean

The following table describes the typographic changes used in this book.

TABLE P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output.	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% You have mail.`
`AaBbCc123`	What you type, contrasted with on-screen computer output.	`machine_name% su` `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value.	To delete a file, type `rm` *filename*.
AaBbCc123	Book titles, new words or terms, or words to be emphasized.	Read Chapter 6 in *User's Guide*. These are called *class* options. You *must* be root to do this.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P-2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

Accessing Sun Documentation Online

The docs.sun.com[sm] web site enables you to access Sun technical documentation on the Web. You can browse the docs.sun.com archive or search for a specific book title or subject at http://docs.sun.com

Also, you can view the online documentation by pointing your browser to the following URL, file:/opt/SUNWconn/em/docs/SEMDOCHP/index.html

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can send your comments by email to docfeedback@sun.com.

Please include the part number of your document in the subject line of your email.

CHAPTER 1

Introduction

This chapter describes the CORBA Gateway at a high level and introduces the CORBA administration tasks.

This chapter describes the following topics:

1.1 What is a CORBA Gateway?

A CORBA Gateway is a set of UNIX processes and shared libraries (see TABLE 1-1) that translate CORBA Manager requests in Interface Definition Language (IDL) to Solstice EM Portable Management Interface (PMI) requests. The CORBA Gateway also translates Solstice EM PMI responses and PMI events to IDL or Internet Inter-ORB Protocol (IIOP) responses and CORBA events.

TABLE 1-1 CORBA Gateway Components

CORBA Component	Type	Description
Request Gateway	UNIX process	Handles initial CORBA client connections to the CORBA Gateway. It also handles the Common Management Information Server (CMIS) requests/responses regarding managed objects.

TABLE 1-1 CORBA Gateway Components *(Continued)*

CORBA Component	Type	Description
Event Gateway	UNIX process	Handles the delivery of CMIS events and notifications to CORBA clients.
Metadata Gateway	UNIX process	Provides CORBA clients with access to ASN1 metadata in Solstice EM.
Event Port Registry	UNIX process	Used by clients to create and register `EventPorts`, in order to get CMIS events and notifications from the Event Gateway.

The SEM CORBA Gateway product can be deployed to work on industry standard ORBs. The ORB's that are currently supported are VisiBroker 4.5 and Orbacus 4.0.5 and Orbix 2000 1.2.1. The product is delivered as a ToolKit, which lets you build the SEM CORBA Gateway for the required ORB, for more information see Chapter 2.

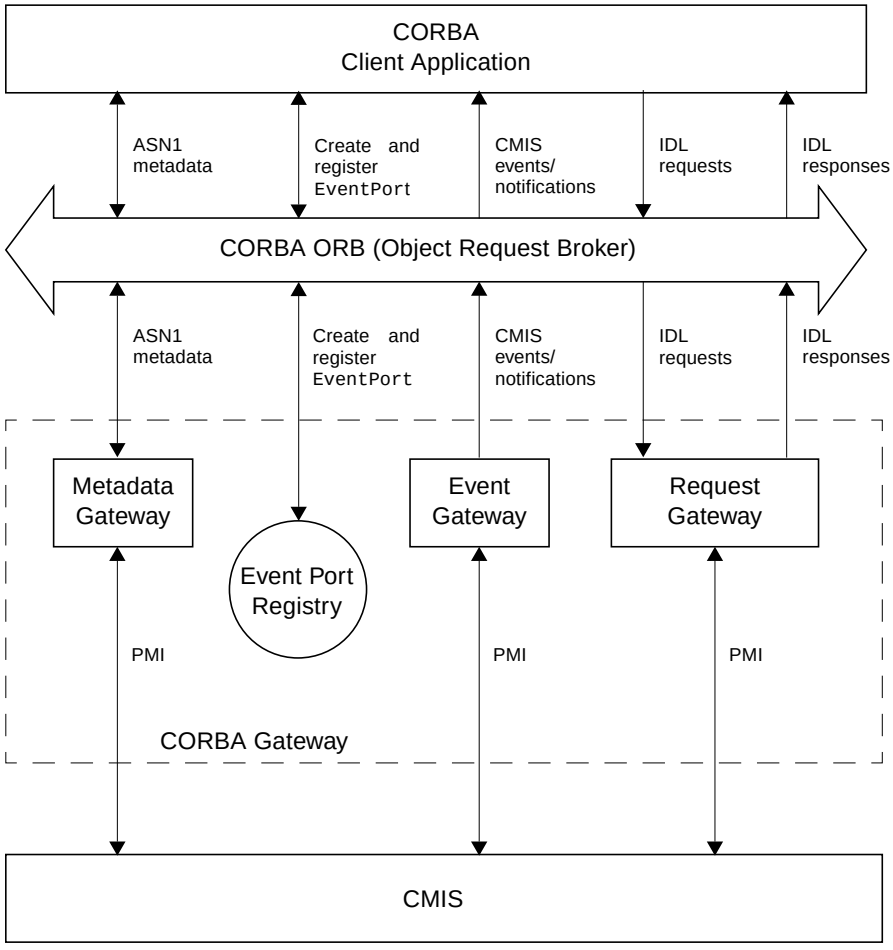

FIGURE 1-1 CORBA Gateway Architecture

1.2 CORBA Gateway Administration

CORBA Gateways administration involves the following:

- **Basic system administration tasks**
  - Configuring the CORBA Gateway to operate smoothly in your particular environment
  - Starting and stopping Gateway processes

  These tasks are described in Chapter 3.
- **ToolKit related tasks**
  - Build the Gateways
  - Make a runtime package
  - Deploy the runtime package on a target machine

  These tasks are described in Chapter 2.
- **Troubleshooting tasks**

  How to find and resolve errors that affect SEM CORBA Gateway operations. These tasks are described in Chapter 3.
- **Performance optimization tasks**
  - Optimizing CORBA Gateway performance
  - Optimizing event performance
  - Optimizing request/response performance

  These tasks are described in Chapter 4.

CHAPTER 2

SEM CORBA ToolKit

The SEM CORBA ToolKit will let you compile, build and package the SEM CORBA Gateway.

This chapter describes the following topics:

2.1 ToolKit Overview

The SEM CORBA ToolKit is a means by which the product delivered is made to be independent of the ORB. It is delivered as a set of libraries, C++ source files, include files, IDL files, Makefiles, and scripts. While installing Solstice EM you are prompted to optionally install the `SUNWemcgs` package which contains all the ToolKit components. All these components (and hence the SEM CORBA ToolKit) are installed during the Solstice EM installation only if you choose to install the `SUNWemcgs` package.

You will need to build the SEM CORBA Gateway by executing the Makefiles provided. The build can be done either individually for each module of the Gateway or all in one shot by executing a `Makefile` in the home directory of the ToolKit.

Note – It is recommended that you run the `Makefile` in the home directory of the ToolKit so that a complete integration of the platform is achieved and the package is created.

2.1.1 Contents of ToolKit

The ToolKit basically consists of components which are ORB dependent and ORB independent. The ORB dependent components are in the form of source files (IDL, C++ and header files). The ORB independent files are distributed as shared libraries. The following table gives the contents and their location in the installation.

TABLE 2-1 ToolKit Contents and Their Location

No.	Component Name	ORB Dependent/ Independent	Location/Name Dependent
1.	IDL files	Dependent	`/opt/SUNWconn/em/src/corba_gateway/idl`
2.	Request Gateway source	Dependent	`/opt/SUNWconn/em/src/corba_gateway/src/request_gw`
3.	Request Gateway shared lib	Independent	`/opt/SUNWconn/em/lib/libem_corba_rgw.so`
4.	Event Gateway source	Dependent	`/opt/SUNWconn/em/src/corba_gateway src/event_gw`
5.	Event Gateway shared lib	Independent	`/opt/SUNWconn/em/lib/libem_corba_eds.so`
6.	Metadata Gateway source	Dependent	`/opt/SUNWconn/em/src/corba_gateway src/metadata_gw`
7.	Metadata Gateway shared lib	Independent	`/opt/SUNWconn/em/lib/libem_corba_mgw.so`
8.	Converter & Utility lib source	Dependent	`/opt/SUNWconn/em/src/lib`
9.	Scripts	Independent	`/opt/SUNWconn/em/src/corba_gateway/scripts`

The `/opt/SUNWconn/em/src/corba_gateway` directory is the home directory of the SEM CORBA ToolKit.

2.1.2 Setting-up Build Environment for SEM CORBA ToolKit

The SEM CORBA ToolKit build environment *needs* to be setup to make it work with the ToolKit.

Setup the build environment by performing the following steps.

1. **Setup Solstice EM environment file by executing the following command:**

   For C shell:

   ```
   source /opt/SUNWconn/em/bin/emenv.csh
   ```

   For K/B shell:

   ```
   . /opt/SUNWconn/em/bin/emenv.sh
   ```

2. **Setup SEM CORBA environment by using the following command:**

   For C shell:

   ```
   source /opt/SUNWconn/em/bin/em_corba_env.csh
   ```

   For K/B shell:

   ```
   . /opt/SUNWconn/em/bin/em_corba_env.sh
   ```

Note – Depending on the ORB selected during installation the `em_corba_env.csh` and `em_corba_env.sh` scripts contain certain environment variables with default values. It is suggested that you modify the default values of the environment variables to your ORB specific parameters.

---

3. **Setup the path for the 5.3 C++ compiler.**

4. **Setup the `EM_BUILDTYPE` environment variable as required, see Section 2.3.1 “Compiling Debug Mode” on page 2-9.**

The following sub-sections will give a description of the individual directories of the ToolKit.

2.1.3 IDL Files

The `/opt/SUNWconn/em/src/corba_gateway/idl` directory has all the information required to build and implement the CORBA interfaces exposed by the SEM CORBA Gateway. The `Makefile` in this directory can be executed to compile the IDL files, compile the C++ files generated and build the libraries. The following table gives a list of the libraries generated based on each of the directories.

TABLE 2-2 ToolKit IDL Directories and Corresponding Libraries Generated

No.	Directory	Library Generated	Target ORB
1.	`auth_proxy`	`libob_idl_auth_proxy.so`	Orbacus
2.	`auth_proxy`	`libvb_idl_auth_proxy.so`	VisiBroker
3.	`auth_proxy`	`libio_idl_auth_proxy.so`	Orbix
4.	`cos`	`libob_idl_cos.so`	Orbacus
5.	`cos`	`libvb_idl_cos.so`	VisiBroker
6.	`cos`	`libio_idl_cos.so`	Orbix
7.	`jidm`	`libob_idl_jidm.so`	Orbacus
8.	`jidm`	`libvb_idl_jidm.so`	VisiBroker
9.	`jidm`	`libio_idl_jidm.so`	Orbix
10.	`jidm_ext`	`libob_idl_jidm_ext.so`	Orbacus
11.	`jidm_ext`	`libvb_idl_jidm_ext.so`	VisiBroker
12.	`jidm_ext`	`libio_idl_jidm_ext.so`	Orbix
13.	`event_gw`	`libob_idl_event_gw.so`	Orbacus
14.	`event_gw`	`libvb_idl_event_gw.so`	VisiBroker
15.	`event_gw`	`libio_idl_event_gw.so`	Orbix
16.	`metadata_gw`	`libob_idl_metadata_gw.so`	Orbacus
17.	`metadata_gw`	`libvb_idl_metadata_gw.so`	VisiBroker
18.	`metadata_gw`	`libio_idl_metadata_gw.so`	Orbix

The installation script installs the `Makefile` required for the ORB. The `Makefile` will have an extension corresponding to the target ORB as follows:

- `Makefile.vb` for VisiBroker
- `Makefile.ob` for Orbacus
- `Makefile.io` for Orbix

The Makefiles compile the IDL files and the corresponding C++ and header files generated are moved to build directories. For example, the `Makefile.vb` (which is the `Makefile` installed for VisiBroker) for the `JIDM` directory will compile all the IDL files; the C++ files generated are copied to `corba_gateway/src/idl_generated/vb/jidm` directory; and the header files are copied to `/opt/SUNWconn/em/include/idl_generated/vb/jidm` directory. The `Makefile.vb` will further compile the C++ files and build the shared libraries (the names of shared libraries are listed in TABLE 2-2) and the libraries are copied to `/opt/SUNWconn/em/lib` directory.

Note – Prior to execution of the Makefiles the SEM CORBA ToolKit build environment should be set (see Section 2.1.2, “Setting-up Build Environment for SEM CORBA ToolKit,” on page 2-3).

The `corba_gateway/src` directory contains the source files which are ORB dependent. These files are compiled using the ORB specific Makefiles that are present in the respective directories.

The source files are distributed in the following directories:

- `request_gw`
- `metadata_gw`
- `event_gw`
  - `corba_eds`
  - `epr`
- `lib`
  - `auth_helper`
  - `cmis_converter`
  - `cppiac`
  - `cgw_utils`
  - `corba_utils`

The `request_gw` has all the ORB dependent C++ source files that need to be compiled to be able to build the SEM CORBA Request Gateway (RGW) and similarly the Metadata Gateway and Event Gateway directories are built using the corresponding Makefiles. The `lib` directory contains the converter and utility libraries that the gateway executables link to.

The following table lists the shared libraries that are compiled and copied from the `lib` directories.

TABLE 2-3 ToolKit Generated Converter and Utility Libraries

No.	Directory	Executable Name	ORB
1.	`lib/auth_helper`	libvb_auth_helper.so	VisiBroker
		libob_auth_helper.so	Orbacus
		libio_corba_utils.so	Orbix
2.	`lib/cmis_converter`	libvb_cmis_converter.so	VisiBroker
		libob_cmis_converter.so	Orbacus
		`libob_cmis_converter.so`	Orbix
3.	`lib/ccpiac`	`libvb_cppiac.so`	VisiBroker
		`libob_cppiac.so`	Orbacus
		`libio_cppiac.so`	Orbix
4.	`lib/cgw_utils`	`libvb_cgw_utils.so`	VisiBroker
		`libob_cgw_utils.so`	Orbacus
		`libio_cgw_utils.so`	Orbix
5.	`lib/corba_utils`	`libvb_corba_utils.so`	VisiBroker
		`libob_corba_utils.so`	Orbacus
		`libio_corba_utils.so`	Orbix

The following table lists the executables that are built and copied to the `/opt/SUNWconn/em/bin` directory.

TABLE 2-4 ToolKit Generated Executables

No.	Directory	Executable Name	ORB
1.	`request_gw`	`em_vb_corba_rgw`	VisiBroker
		`em_ob_corba_rgw`	Orbacus
		`em_io_corba_rgw`	Orbix
2.	`metadata_gw`	`em_vb_corba_mgw`	VisiBroker
		`em_ob_corba_mgw`	Orbacus
		`em_io_corba_mgw`	Orbix
3.	`event_gw/epr`	`em_vb_corba_epr`	VisiBroker

TABLE 2-4 ToolKit Generated Executables *(Continued)*

No.	Directory	Executable Name	ORB
		em_ob_corba_epr	Orbacus
		em_io_corba_epr	Orbix
4.	event_gw/corba_eds	em_vb_corba_eds	VisiBroker
		em_ob_corba_eds	Orbacus
		em_io_corba_eds	Orbix

2.2 Building the Executables

The SEM CORBA ToolKit can build executables for any one of the three ORB that are supported.

Build the executables by performing the following steps:

1. **Login as root.**
2. **Setup the SEM CORBA ToolKit development environment (see, Section 2.1.2, "Setting-up Build Environment for SEM CORBA ToolKit," on page 2-3).**

   By performing this step you would have set the path to the ORB, the Make utility and the C++ compiler; also you would have set the LD_LIBRARY_PATH for the build commands.
3. **Execute the following command from home directory of the ToolKit, to set some of the options and environment variables to be used in the Makefiles.**

```
make firstmake
```

4. **Execute the following command if you have a previously installed ToolKit.**

```
make clean
```

   You may skip this step if you do not have a ToolKit previously installed.
5. **Execute the following command from the ToolKit home directory to do a complete build and installation of the SEM CORBA Gateway.**

```
make install
```

6. **Execute the following command to create the corba-runtime package** `SUNWemcgr`

```
make package
```

The `SUNWemcgr` package is created in the ToolKit home directory of the development machine.

2.3 Installing Runtime Package on Target Machine

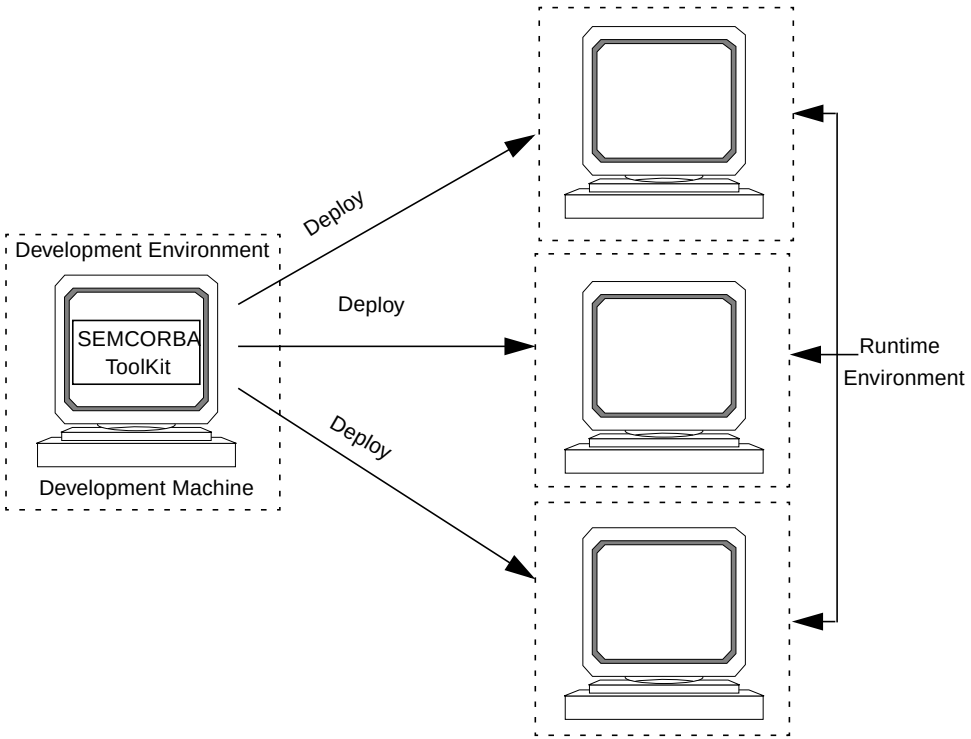

FIGURE 2-1 SEM CORBA ToolKit Development Environment

If you want to install the runtime package on any other target machine other than the development machine perform the following steps:

1. **Log on to the target machine as root.**
2. **Copy** `SUNWemcgr` **package from the development machine to the target machine.**
3. **Execute the following command in the target machine:**

```
/usr/sbin/pkgadd -d. SUNWemcgr
```

2.3.1 Compiling Debug Mode

The SEM CORBA Gateway can be built in the debug mode by using the environment variable `EM_BUILDTYPE`. Under the normal circumstances, this variable is set to `OPT`. To build the SEM CORBA Gateway in debug mode, this environment variable should be set to `DEBUG`. Use the following command to set debug mode.

In C shell:

```
setenv EM_BUILDTYPE DEBUG
```

In K/B shell:

```
EM_BUILDTYPE=DEBUG;export EM_BUILDTYPE
```

Once `EM_BUILDTYPE` is set, the execution of `Makefile` will compile and build the executables and libraries in debug mode.

2.4 Other SEM CORBA ToolKit Directories

The SEM CORBA ToolKit home has the following directories apart from the `src` and `idl` directories discussed previously.

- `build_config`
- `scripts`

The `build_config` contains the `Master.Makefile` which gets included in all the other `Makefile`s and the template file for the ORB (`vb.tmpl` for VisiBroker; `ob.tmpl` for Orbacus; `io.tmpl` for Orbix).

The `scripts` directory contains the shell scripts that define the environment variables required to be able to use the ToolKit.

CHAPTER 3

Administering the System

This chapter describes how to configure the SEM CORBA Gateway as well as how to start, stop, and troubleshoot CORBA Gateway processes.

This chapter describes the following topics:

3.1 Configuring SEM CORBA Gateway

To configure SEM CORBA Gateway, you must configure its individual component gateways. You can configure these gateways by modifying their corresponding configuration files (see TABLE 3-1) and restarting the SEM CORBA Gateway to reflect the changes. By default, the configuration files are located at `/var/opt/SUNWconn/em/conf` directory.

TABLE 3-1 SEM CORBA Gateway Configuration Files

File	Description
`em_corba_epr.conf`	Configuration file for Event Port Registry
`em_corba_rgw.conf`	Configuration file for Request Gateway
`em_corba_eds.conf`	Configuration file for Event Distribution Server
`em_corba_mgw.conf`	Configuration file for Metadata Gateway

To modify a configuration file, set the values of its environment variable to the appropriate values if different from the default values. TABLE 3-2 describes the SEM CORBA Gateway configuration variables.

TABLE 3-2 SEM CORBA Gateway Configuration Variables

Variable	Description
EM_MIS_DEFAULT_HOST	Name of MIS to which SEM CORBA Gateway connects.
EM_CORBA_EPR_LOG_FILE	The file used by Event Port Registry to log messages.
EM_CORBA_EPR_DEBUG_PORT	Debug port for remote dynamic debugging of Event Port Registry.
EM_CORBA_RGW_LOG_FILE	The file used by Request Gateway to log messages.
EM_CORBA_RGW_DEBUG_PORT	Debug port for remote dynamic debugging of Request Gateway.
EM_CORBA_RGW_MAX_POOL_THREAD	Shared pool sizes for reply thread pools and other thread pools.
EM_CORBA_MGW_LOG_FILE	The file used by Metadata Gateway to log messages.
EM_CORBA_MGW_DEBUG_PORT	Debug port for remote dynamic debugging of Metadata Gateway.

Note – If you run `em_services` with the `-reload` or `-r` option after you modify the configuration files, your modifications will be lost.

To make your changes permanent, modify and save the relevant configuration file in the directory `/opt/SUNWconn/em/build/acct` (assuming you installed Solstice EM in the default `/opt` location).

Caution – Be careful with permanent modifications. You may lose original default values unless you have a backup copy.

3.1.1 Start-up Options for SEM CORBA Gateway

The SEM CORBA Gateway requires the following command line arguments during start-up:

For VisiBroker:

```
-ORBInitRef NameService=iioploc://$SYSTEMID:$EM_CNS_PORT/
NameService
```

The SYSTEMID and EM_CNS_PORT environment variables specify the system on which naming services are to be located and the port through which the naming services will be contacted. The SYSTEMID variable is defined in the em_cgw_services script file and the EM_CNS_PORT variable is defined in the em_corba_env.csh and em_corba_env.sh files.

For Orbacus:

```
-ORBservice NameService corbaloc::$SYSTEMID:$EM_CNS_PORT/
NameService
```

The SYSTEMID and EM_CNS_PORT environment variables specify the system on which naming services are to be located and the port through which the naming services will be contacted. The SYSTEMID variable is defined in the em_cgw_services script file and the EM_CNS_PORT variable is defined in the em_corba_env.csh and em_corba_env.sh files.

For Orbix:

```
-ORBdomain_name ${IT_DOMAIN_NAME} -ORBconfig_domains_dir
${IT_CONFIG_DOMAINS_DIR}
```

The IT_DOMAIN_NAME environment variable specifies the configuration domain name to use. The IT_CONFIG_DOMAINS_DIR environment variable specifies the directory containing the configuration file *DomainName*.cfg.

Generally the SEM CORBA Gateway services are started, using em_cgw_services, which is invoked by the following command:

```
em_cgw_services -start
```

This command will also try to start the *naming service* on the host machine. This will work fine for VisiBroker and Orbacus. In case of Orbix, to be able to run naming service, the host machine should have an Orbix server runtime environment. If the Orbix sever runtime environment is setup on a machine other than the host machine and naming service is started on that machine, it is possible for SEM CORBA Gateway services to use the naming service. In such a case, you will have to suitably modify the `em_cgw_services` script so as to not start or stop the Orbix services.

Note – The Orbix client runtime environment should be installed on the machine on which you want to deploy the Orbix client applications.

3.1.2 Setting the Default Root Naming Context

The default root naming context is specified by the environment variable `EM_CORBA_ROOT_NAMING_CONTEXT` and the default value is the hostname where `em_cgw_services` is run. You can use this variable to distribute the ORB and `CosNamingExtFactory` to a machine different from the one on which the SEM CORBA Gateway is running.

3.1.3 Configuring the CORBA Event Port Registry

TABLE 3-3 describes the configuration of CORBA Event Port Registry (EPR), while CODE EXAMPLE 3-1 shows a sample EPR configuration file.

TABLE 3-3 CORBA EPR Configuration

Action	Procedure
To log errors or output messages from the Event Port Registry	Set the `EM_CORBA_EPR_LOG_FILE` environment variable in the `em_corba_epr.conf` configuration file to the name of the log file to use. The default value of the variable is `/var/opt/SUNWconn/em/debug/em_corba_epr.log`
To configure the debug port for troubleshooting	Set the `EM_CORBA_EPR_DEBUG_PORT` environment variable in `em_corba_epr.conf` to the debug port. The default value is `6660`. For more information on troubleshooting, See Section 3.3, “Troubleshooting SEM CORBA Gateway Processes,” on page 3-10.

CODE EXAMPLE 3-1 Sample EPR Configuration File

```
# Copyright 06/01/99 Sun Microsystems, Inc. All Rights Reserved.

#pragma ident  "@(#)em_corba_epr.conf   1.1 99/06/01 Sun
Microsystems"

# CORBA event gateway log file in full path
EM_CORBA_EPR_LOG_FILE:           /var/opt/SUNWconn/em/debug/
em_corba_epr.log

# CORBA event gateway debug port
EM_CORBA_EPR_DEBUG_PORT:         6660
```

3.1.4 Configuring the CORBA Request Gateway

TABLE 3-4 describes the configuration of CORBA Request Gateway (RGW), while shows a sample RGW configuration file.

TABLE 3-4 CORBA RGW Configuration

Action	Description
To specify the MIS to which the CORBA RGW connects	Set the `EM_MIS_DEFAULT_HOST` variable in the `em_corba_rgw.conf` configuration file to the name of the MIS. The default value is `localhost`.
To log errors or output messages from the CORBA RGW	Set the `EM_CORBA_RGW_LOG_FILE` environment variable in `em_corba_rgw.conf` to the name of the log file to use. The default value of the variable is `/var/opt/SUNWconn/em/debug/em_corba_rgw.log`
To configure the debug port for troubleshooting	Set the `EM_CORBA_RGW_DEBUG_PORT` environment variable in `em_corba_rgw.conf` to the debug port. The default value is `6666`. For more information on troubleshooting, refer to Section 3.3, “Troubleshooting SEM CORBA Gateway Processes,” on page 3-10.
To modify the maximum number of threads (thread-pool size) that the CORBA RGW uses if the hardware configuration is different from the reference configuration	Set the `EM_CORBA_RGW_MAX_POOL_THREADS` environment variable in `em_corba_rgw.conf` to the maximum number of threads. The default value is `3`.

CODE EXAMPLE 3-2 Sample RGW Configuration File

```
# Copyright 06/01/99 Sun Microsystems, Inc. All Rights Reserved.

#pragma ident  "@(#)em_corba_rgw.conf   1.1 99/06/01 Sun
Microsystems"

# MIS default name
EM_MIS_DEFAULT_HOST:             localhost

# CORBA request gateway log file in full path
EM_CORBA_RGW_LOG_FILE:           /var/opt/SUNWconn/em/debug/
em_corba_rgw.log

# CORBA request gateway maximum pool threads
EM_CORBA_RGW_MAX_POOL_THREADS:   3

# CORBA request gateway debug port
EM_CORBA_RGW_DEBUG_PORT:         6666
```

3.1.5 Configuring the CORBA Event Distribution Server

As it is possible to start more than one CORBA Event Distribution Server (EDS) as part of a given CORBA Gateway, and because it is hard to map configuration file variables to instances of CORBA EDS, the CORBA EDS does not use configuration files to configure itself. Instead, CORBA EDS relies on the command-line arguments shown in TABLE 3-5 for configuration.

TABLE 3-5 CORBA EDS Configuration Commands

Command	Description
`mis`	Specifies the MIS from which events should be gathered.
`port`	Specifies the debug port.
`log`	Specifies the Filename where error or output messages are logged.
`threads`	Specifies the Thread-pool size used by CORBA EDS instances.

By default, `em_cgw_services` starts two instances of CORBA EDS. To start more or fewer instances, modify the `start_corba_event_gw()` function in the `em_cgw_services` script located in the `/opt/SUNWconn/bin` directory.

CODE EXAMPLE 3-3 shows the default contents of this script, and CODE EXAMPLE 3-4 shows a modified version to start three instances of CORBA EDS.

Note – The CORBA EDS 1 default debug port is 6500. The CORBA EDS 2 default debug port is 6502.

CODE EXAMPLE 3-3 Default Contents of the Script `em_cgw_services`[1]

```
start_corba_event_gw() {
    start_corba_eds em_corba_eds1.log 6500 3
    debugmsg "Started the first CORBA EDS"

    start_corba_eds em_corba_eds2.log 6502 3
    debugmsg "Started the second CORBA EDS"
}
```

1. The last argument in the `start_corba_eds` command is the maximum number of threads in the thread pool to use. A non- zero value should be specified for this argument.

CODE EXAMPLE 3-4 Modified `em_cgw_services` Script to Start Up 3 EDS Gateways

```
start_corba_event_gw() {
    start_corba_eds em_corba_eds1.log 6500 3
    debugmsg "Started the first CORBA EDS"

    start_corba_eds em_corba_eds2.log 6502 3
    debugmsg "Started the second CORBA EDS"

    start_corba_eds em_corba_eds3.log 6503 3
    debugmsg "Started the third CORBA EDS"
}
```

3.1.6 Configuring the CORBA Metadata Gateway

TABLE 3-6 describes the configuration of CORBA Metadata Gateway (MGW), while CODE EXAMPLE 3-5 shows a sample MGW configuration file.

TABLE 3-6 CORBA MGW Configuration

Action	Description
To specify the MIS to which the CORBA MGW connects	Set the `EM_MIS_DEFAULT_HOST` variable in the `em_corba_mgw.conf` configuration file to the name of the MIS. The default value is `localhost`.
To log errors or output messages from the gateway	Set the `EM_CORBA_MGW_LOG_FILE` environment variable in `em_corba_mgw.conf` to the name of the log file to use. The default value of is `/var/opt/SUNWconn/em/debug/em_corba_mgw.log`
To configure the debugging port for troubleshooting	Set the `EM_CORBA_MGW_DEBUG_PORT` environment variable in `em_corba_mgw.conf` to the debug port. The default value is `6664`. For more information on troubleshooting, refer to Section 3.3, “Troubleshooting SEM CORBA Gateway Processes,” on page 3-10.

CODE EXAMPLE 3-5 Sample MGW Configuration File

```
copyright 06/01/99 Sun Microsystems, Inc. All Rights Reserved.
#pragma ident "@(#)em_corba_mgw.conf 1.1 99/06/01 Sun Microsystems"

# MIS default name
EM_MIS_DEFAULT_HOST:             localhost

# CORBA metadata gateway log file in full path
EM_CORBA_MGW_LOG_FILE:           /var/opt/SUNWconn/em/debug/
em_corba_mgw.log

# CORBA metadata gateway debug port
EM_CORBA_MGW_DEBUG_PORT:         6664
```

3.1.7 Configuring Authentication Modules

You can use the CORBA Gateway to develop and use your own authentication modules. For information on how to develop your own authentication modules, refer to Chapter 2 “Interacting With SEM CORBA Gateway” in *Developing CORBA Applications*.

To use your own authentication module (library), do the following:

1. **Stop the CORBA Gateway.**

```
em_cgw_services -stop
```

2. **Replace the existing authentication library** `libauth_server.so.1` **on the machine where the CORBA Gateway is running, with your authentication library.**

3. **Restart the CORBA Gateway.**

```
em_cgw_services -start
```

3.2 Starting and Stopping CORBA Gateway Processes

To start or stop CORBA Gateway processes, use one of the following commands:

- `em_services` – Automatically starts or stops Solstice EM including CORBA Gateway processes.
- `em_cgw_services` – Automatically starts or stops the individual gateway components and ORB-related processes only, *not* the whole Solstice EM. This command starts one instance of each individual gateway and two instances of CORBA EDS.

Note – For Orbix 2000, it is assumed that the you have set up a configuration repository domain with *atleast* the *naming service*. The Solstice EM script `em_cgw_services` uses the scripts `start_`*DomainName*`_services` and `stop_`*DomainName*`_services` for starting and stopping the Orbix services. If the `start_`*DomainName*`_services` and `stop_`*DomainName*`_services` are not installed in the standard location (*InstallationDirectory*`/bin`) then their current location should be indicated in the `em_cgw_services` script.

3.3 Troubleshooting SEM CORBA Gateway Processes

You can troubleshoot CORBA Gateway processes as follows:

- Check the log files associated with SEM CORBA Gateway processes.
- Use the `em_debug` command to turn on dynamic debugging in a CORBA Gateway.

3.3.1 Checking the Log Files

When troubleshooting SEM CORBA Gateways, first check the log files associated with the individual SEM CORBA Gateway processes. These files contain the error messages that are logged by the Gateway. TABLE 3-7 describes the log files that are generated by default, by the CORBA Gateway. These files are present in the `/var/opt/SUNWconn/em/debug` directory.

TABLE 3-7 CORBA Gateway Log Files

File	Description
`em_corba_epr.log`	EPR log file
`em_corba_rgw.log`	RGW log file
`em_corba_mgw.log`	MGW log file
`em_corba_eds1.log`	CORBA EDS 1 log file
`em_corba_eds2.log`	CORBA EDS 2 log file

You can specify different files to be used as log files by changing the values of the log file configuration variables (see Section 3.1, “Configuring SEM CORBA Gateway,” on page 3-1 for more information).

CODE EXAMPLE 3-6 Sample Log File Contents for the Request Gateway[1]

```
rgw_trace: RequestSAPManager::RequestSAPManagerrgw_debug:
Successfully created User SAP
rgw_debug: Started the thread safe PMI scheduler
rgw_debug: Obtained initial name service reference
rgw_trace: JIDM::ProxyAgentController - created
rgw_debug: JIDM::ProxyAgentController is ready
rgw_debug: JIDM::ProxyAgentFinder is ready
```

CODE EXAMPLE 3-6 Sample Log File Contents for the Request Gateway[1] *(Continued)*

```
rgw_trace: RequestSAPManager::RequestSAPManagerrgw_debug:
Successfully created User SAP
rgw_debug: SEM::AuthenticationProxy is ready
rgw_debug: Ready to accept client requests
rgw_trace: ProxyAgentFinderImpl::access_domain() starts
rgw_trace: JIDM::ProxyAgentFinder - validating key and access
criteria
rgw_trace: JIDM::ProxyAgentFinder - authenticating user profile in
[faith MIS]
rgw_trace: ProxyAgentFinderImpl::find_matching_proxy_agent()
starts
rgw_trace: JIDMProxyAgentImpl::JIDMProxyAgentImpl() start
rgw_trace: JIDMProxyAgentImpl::JIDMProxyAgentImpl() end
rgw_debug: OSIMgmtExt::ProxyAgent is ready
rgw_trace: ProxyAgentFinderImpl::returning this - access_domain()
ends
rgw_trace: JIDMProxyAgentImpl::access_criteria() start
rgw_trace: GetPendingRequest::translate_request() start
rgw_debug: message type = get request
rgw_debug: id = 0
rgw_debug: source =
rgw_debug:   aclass = PRIM, atag = 2
  aval =
"[0xff][0xff][0x2][0xe3][0xc4][0x1][0x4][0x81][0x9e][0xe6][0xa3]
"
rgw_debug: dest =
rgw_debug:   aclass = PRIM, atag = 2
  aval =
"[0xff][0xff][0x2][0x15][0xb3][0x1][0x4][0x81][0x9e][0xe6][0xa3]
"
rgw_debug: remote =
rgw_debug:   aclass = DEF, atag = 0
  aval = Du: no data unit allocated
rgw_debug: mode = CONFIRMED
rgw_debug: app_context = Du: no data unit allocated
rgw_debug: oc =
rgw_debug:  Tag  Len Value
```

1. In the sample above, the `em_debug` is used to turn on dynamic debugging for the Request Gateway.

3.3.2 Using Dynamic Debugging

You may occasionally need to use the `em_debug` command to turn on dynamic debugging for SEM CORBA Gateway. The output of `em_debug` is particularly useful if you have to file a bug against the SEM CORBA Gateway. TABLE 3-8 describes the debug objects that you can enable:

TABLE 3-8 CORBA Gateway Debugging Objects

Error Objects	Debug Objects	Corresponding Gateway
`rgw_error`	`rgw_debug`	RGW
`mgw_error`	`mgw_debug`	MGW
`epr_error`	`epr_debug`	EPR
`egw_error`	`egw_debug`	CORBA EDS

The debug objects[1] print out extensive debug statements, which will involve printing the contents of a message going back and forth between SEM CORBA Gateway and MIS. The debug statements are not actual error indicators but are a trace of events that occur leading to the error.

For example, to see the error messages from the RGW, you can enter the following command:

```
em_debug -port 6666 -c 'on rgw_error'
```

The `em_debug` command sends messages to the window in which the command is executed. If you expect a large number of messages to be generated, you could redirect the output of the command to a file which you can view with a text-editor or by executing the `tail -f` command.

1. Enabling debug objects degrades the performance of the system because of the large number of messages that get generated. It is recommended that you disable debug objects when you no longer need the debug information.

CHAPTER 4

Improving CORBA Gateway Performance

This chapter describes how to improve the performance of CORBA Gateways.

This chapter describes the following topics:

4.1 Optimizing CORBA Gateway Performance

Optimizing the CORBA Gateway depends very much on the configuration parameters you specify (see TABLE 3-2). The most critical of the parameters is the thread pool size. Too many threads affect the over-all performance of Solstice EM, while too few threads result in the application requests getting timed out. Other configuration parameters such as `EM_MIS_HOST_NAME`, facilitate running of CORBA Gateways on different machines in order to enhance performance.

To improve performance across the whole of Solstice EM, you should install CORBA Gateway on a machine different from the one where the MIS is installed.

To optimize the performance of events only, see Section 4.2, “Optimizing Event Performance,” on page 4-2 for more information. To optimize the performance of management requests/responses only, see Section 4.3, “Optimizing Request/Response Performance,” on page 4-2 for more information.

4.2 Optimizing Event Performance

You can optimize event performance in the following three ways:

- You can run the CORBA Event Distribution Server on different machines to distribute the load when the incoming event rate is high.
- You can have your own CORBA Event Channel, which connects to the SEM CORBA Event Gateway and distributes the events to the clients. This method will reduce the load on the Solstice EM Event Gateway, for obvious reasons. The access control in such a case will be done only on the CORBA Event Channel which connects to the Solstice EM Event Gateway and hence may have a diluting effect.
- You can find the optimal thread pool size for CORBA Event Distribution Server by gradually increasing the size of the thread pool (using the `-threads` option for the CORBA Event Distribution Server) until you identify the optimal size for your configuration.

4.3 Optimizing Request/Response Performance

You can optimize request/response performance in the following two ways:

- You can run the Request Gateway on different machines to distribute the load.
- You can find the optimal thread pool size for the CORBA Request Gateway by gradually increasing the value of the `EM_CORBA_RGW_MAX_POOL_THREADS` environment variable until you identify the optimal size for your configuration.

Index

D

E

G

S

T

V