APPENDIX A

Advanced Usage

Overview of Component Servers

The Cisco IP Manager software consists of several executable server files that can be run concurrently or individually, depending on what features you need. These servers can be installed on multiple hosts in any combination, or all on one host. The Cisco IP Manager software is not dependent on any one server being located on a particular host. However, SQL*Net must be installed whenever any server other than INGServer or CNGSServer (*ding* package) is installed on a host that does not contain the Oracle database.

While it is possible to start the servers from the command line, it is strongly recommended that you always use the *ipmgr.putit* and *ipmgr.launch* scripts described in the chapter “Getting Started.” These files will source the file *ipmgr.launch.csh* to set up certain environment variables that the servers need, and they define a common set of server launch flag settings. You can change these flags by editing the file *ipmgr.launch.csh*.

In order to use these scripts, you must also run the utility *ipmgr.configure*, as described in the “Getting Started” chapter.

The startup scripts allow you to launch all of the servers at once (if they are all installed on a single host), or individual servers one at a time.

If you start the servers individually, NS and ES must be launched first, then LOGServer, then NAMServer, then the others in any order.

Note If ES (the Event Server) stops for any reason, you must shut down, unregister, and restart *all* of the servers that you use. The *ipmgr.rmit* utility unregisters the servers from the Orbix Naming Service (NS).

Though it is not recommended that you do so for normal use, it is possible to start most of the servers from the command line. If you launch a server without any arguments, even if the server is already running, a usage-text message will define the various launch flags for that server (except ES and NS, which fail and write core files if the Orbix daemon is not running, and INGServer, which will hang).

If you launch a server (except NS, ES, and CNGSServer) with the **-V** flag, the version number of that server will be output to **stdout**.

Directory Structure

When you install any of the server packages as described in the chapter "Getting Started," the following subdirectory structure will be created in the installation directory:

```
SRVRS
   pack
      OrbixMT
      OrbixName
   ps
      bin
      expect
      nsm-agent
      utility
```

The *OrbixMT* subdirectory contains the Orbix (CORBA) operating files. The Naming Server (NS) is located in the *SRVRS/pack/OrbixName/1.1/bin* subdirectory.

The *ps* subdirectory contains the Cisco IP Manager operating files.

The servers AASServer, AUTHServer, CTMServer, ES, INGServer, LOGServer, NAMServer, and NEMServer are located in appropriately named subdirectories under the *SRVRS/ps/bin* subdirectory of your Cisco IP Manager installation.

The CNGSServer is located in the *SRVRS/ps/nsm-agent/bin/solaris* subdirectory.

Installation of the GUI package will create the following subdirectory structure in the installation directory:

```
GUI
   OrbixMT
   java
   ps_java_env
```

The *GUI/OrbixMT* subdirectory contains another set of Orbix operating files. The *java* subdirectory contains the operating files needed to create the user interface. The *ps_java_env* subdirectory contains additional files needed by the GUI application.

The Orbix files are included in both GUI and server installations to accommodate installation of remote GUI workstations, independent of server installations. However, if GUI and server packages are installed on the same machine and you are not using the Cisco-provided launch scripts, only one set of Orbix files is needed. You may delete the other set.

However, if you do so, be aware of the following:

- The scripts *ipmgr.putit* and *ipmgr.launch* will look in the *SRVRS* subdirectory for the Orbix daemon when the servers are launched.
- When you source the file *ipmgr.launch.csh*, the *SRVRS* Orbix installation is added to your PATH enviornment variable. If you delete this set of files, you will not be able to launch the Orbix utilities from a different subdirectory.
- If the Orbix daemon is not already running when you launch the GUI using the *ipmgr.gui* launch script, the script will look for the daemon in the *GUI* subdirectory.

Orbix

The Orbix daemon must be launched before you can register any of the Cisco IP Manager servers. It must also be running when you launch the GUI application.

If you use any of the launch scripts described in the chapter “Getting Started,” the Orbix daemon will be started for you. If you do not use these scripts, you must start the daemon yourself, as follows:

```
orbixd -c $IPMGR_ORBIX_HOME/orbixd.log &
```

The **-c** parameter creates an Orbix checkpoint file named *orbixd.log* in the directory named by the IPMGR_ORBIX_HOME environment variable, which was set when you sourced *ipmgr.csh*. This checkpoint file allows a new daemon to recover data about the Cisco IP Manager servers in the event of an unexpected termination of the Orbix daemon.

The Orbix daemon is located in the *OrbixMT/2.3/bin* subdirectory.

Remote Login

To run the GUI remotely (via the **rlogin** command), you must first invoke the UNIX command **xhost + <remote_machine>** on the machine where you are working, and once you have logged in to the remote machine, the environment variable DISPLAY must be set to your local machine's hostname:

```
xhost +<remote machine>
rlogin <remote machine>
setenv DISPLAY <your machine name>:0
```

However, the machine on which the process is actually run must have sufficient resources to handle all of the concurrent sessions without terminating and writing a core file.

Server Launch Options

Servers can be launched in a variety of configurations, by setting their launch flags as described in the following sections. (If you launch a server executable with the **-V** flag—whether the server is running or not—a message will be sent to **stdout** identifying the version of that server. If you specify the **-V** flag when the server is not already running, the message will be generated, but the server will not be launched.)

AASServer (Authorization and Access Server)

The Authorization and Access Server provides capability for protecting Cisco IP Manager information from unauthorized manipulation. The server manages administration information such as user groups and access permission, and it provides services to enforce access control on the managed resource. Every Cisco IP Manager operation can be validated against the permission which defines what a user can do on a particular resource. The server handles creation, deletion, and modification of permissions and user groups; addition and deletion of user group membership; and validation of domain and system-based resources.

Permissions will be checked for those operations requiring user permission by CTMServer, LOGServer, NAMServer and NEMServer only if those servers are launched with the **-A** flag (the default condition in the *ipmgr.launch* script).

AASServer Command line options

Required

```
-u <database user name>
-p <database password>
```

Optional

```
-n <database name>
-s <database server name>

   With an Oracle database, both -n and -s flags are equivalent to
   $ORACLE_SID

-L <Cisco IP Manager log output>

   Can be server, cout, console, or <filename>. If server, then output
   will be sent to the LOGServer for storage in the database. If -L is
   not specified, no log messages will be generated.

-q <thread pool size> (defaults to 5)
-i <database min. connections> (defaults to 1)
-V // launch server executable with the -V flag to see version number
-x <database max. connections> (defaults to 5)
```

AUTHServer (Authentication Server)

The Authentication Server provides authentication services for the Cisco IP Manager software. It manages user information—user name, password, and profile—and provides login verification.

AUTHServer Command line options

Required

```
-u <database user name>
-p <database password>
```

Optional

```
-L <Cisco IP Manager log output>

   Can be server, cout, console, or <filename>. If server, then output
   will be sent to the LOGServer for storage in the database. If -L is
   not specified, no log messages will be generated.

-n <database name>
-q <thread pool size> (default is 5)
-s <remote database server name>

   With an Oracle database, both -n and -s flags are equivalent to
   $ORACLE_SID

-V // launch server executable with the -V flag to see version number
```

CTMServer (Configuration Template Manager)

The Configuration Template Manager (CTMServer) is responsible for configuration of templates and dynamic data management. The CTMServer provides the following services to the Cisco IP Manager software: creation, deletion and modification of configuration templates and dynamic data; and generation of configuration files by merging template and data files.

CTMServer Command line options

Required

```
-u <database user name>
-p <database password>
```

Optional

```
-A // launch with -A to enable Access Check for Domain Model Templates
-i <database min. connections> (default is 1)
-L <Cisco IP Manager log output>

   Can be server, cout, console, or <filename>. If server, then output
   will be sent to the LOGServer for storage in the database. If -L is
   not specified, no log messages will be generated.

-N <NAMServer host name>
-n <database name>
-q <thread pool size> (defaults to 10)
-s <remote database server name>

   With an Oracle database, both -n and -s flags are equivalent to
   $ORACLE_SID

-T <AAS Server host name> // not currently implemented
-V // launch server executable with the -V flag to see version number
-x <database max. connections> (default is 5)
```

CNGSServer

The CNGSServer generates Integrity Check, Syntax Check, and Unconnected WAN Interface reports. Each time a report is generated, the server creates new subdirectories in */tmp*. To avoid running out of disk resources, you should delete these directories when you are finished with the reports.

With one exception, these reports will always be displayed in a Java-based window generated by the CNGSServer. The exception is the Integrity Checks *diff* report generated through the API function **diffIntegrityCheckInHTML()**, in the ING Module. The full path to this report, in HTML format, will be constructed as follows:

```
<base_directory>/<server><n>/<user>/<internal_identifier>/html_reports/
     integrity_checks_report/index.html
```

The value of **base_directory** is the string specified in the **-htmlDirectory** argument at the launch of the CNGSServer. The value of **server** is the string specified in the **-serverName** argument. The **n** following the server name is the process ID of the CNGSServer which generated the report. The value of **user** is the UNIX user ID (not the Cisco IP Manager user ID). The value of **internal_identifier** is a number used internally by the *apply_configlets* script to distinguish itself from other applications.

For example, if you are logged in to your machine as *UNIX_User*, the CNGSServer was launched with /var/*myDirectory* and *CNGS* as arguments, the CNGS host machine runs the server under process ID 3216, and the script's internal identifier is 499517411, then the HTML version of the report will be saved as:

```
/var/myDirectory/CNGS3216/UNIX_User/499517411/html_reports/
   integrity_checks_report/index.html
```

CNGSServer Command Line Options

Optional

```
-htmlDirectory  base directory used in the construction of paths to
                HTML reports (default is /tmp)
-serverName     name of server (default is CNGSServer)
```

ES (Event Service)

The Event Service (ES) provides an abstraction layer which gives the application the capability of pushing creation, deletion, and modification events. When launching this server from the command line, you must specify event channels.

This server is launched by the *ipmgr.launch* utility with the following command:

```
es_1.0_patched Domain_Creation Domain_Modification Domain_Deletion
Permission_Creation Permission_Modification Permission_Deletion
Usergroup_Creation Usergroup_Modification Usergroup_Deletion
User_Creation User_Deletion User_Modification Device_Creation
Device_Modification_Comm Device_Modification_Config
Device_Modification_Admin Device_Deletion Template_Creation
Template_Modification Template_Deletion -nonames
```

INGServer

The IP Manager-Netsys Gateway (ING) Server is the gateway to the NSM-Agent Server, which provides configuration-file checking capability. ING Server supports multiple configuration file and network integrity checking requests simultaneously.

INGServer Command line options

Optional

```
-V // launch server executable with the -V flag to see version number
```

LOGServer (Log and Audit Server)

The Log and Audit server (LOGServer) manages the logging of messages from the component servers. These messages contain date and time of the operation (GMT), the name of the server which generated the message, location in the program, a message category identifier, a message description, and a user name. Output can be captured into the database or a user-specified file, or redirected to the console or terminal. LOGServer is itself capable of generating messages that can be captured in the database.

LOGServer Command line options

Required

```
-u <database user name>
-p <database password>
```

Optional

```
-A // launch with -A to use Authorization and Access Control
-L <Cisco IP Manager log output>

   Can be server, cout, console, or <filename>. If server, then output
   will be stored in the database. If -L is not specified, no log
   messages will be generated.

-n <database name>
-q <thread pool size> (default is 5)
-s <database server name>

   With an Oracle database, both -n and -s flags are equivalent to
   $ORACLE_SID

-V // launch server executable with the -V flag to see version number
```

NAMServer (Network Administration Manager)

The Network Administration Manager (NAMServer) provides the capabilities needed to manage the administration of domains and network elements. These capabilities include creation, deletion, and modification of domains and routers; location of objects by name-lookup; and search by predefined filtering criteria.

NAMServer Command line options

Required

```
-u <database user name>
-p <database password>
```

Optional

```
-A // launch with -A to use Authorization and Access Control
-n <database name>
-L <Cisco IP Manager log output>

   Can be server, cout, console, or <filename>. If server, then output
   will be sent to the LOGServer for storage in the database. If -L is
   not specified, no log messages will be generated.

-q <thread pool size> (defaults to 5)
-s <database server name>

   With an Oracle database, both -n and -s flags are equivalent to
   $ORACLE_SID

-V // launch server executable with the -V flag to see version number
```

NEMServer (Network Element Manager)

The Network Element Manager (NEMServer) is responsible for managing network elements and for communications between elements and servers. NEMServer maintains communication status, supports network element operations, and monitors element status. NEMServer supports downloading and uploading configuration files.

For each operation, NEMServer returns an operation error code which is written to a system log file, with detailed information identifying the operation and its result, the network element, and the time.

TFTP

NEMServer was designed to exchange data with network elements using a TFTP server, which you specify using the **-f** option.

If this option is not specified, data will be downloaded to devices line by line using the IOS **configure terminal** command and will be uploaded from devices by capturing the output following a **show** command. (However, this will result in a slow data exchange rate.)

It is recommended that each machine on which you install NEMServer be configured to act as its own TFTP server. To do this, you must specify either its host name or IP address using the **-f** option when the server is launched. (If you use hostnames, every device that communicates through the TFTP server must be able to resolve names.) Be sure everyone who will be running the Cisco IP Manager software has **read/write** access to the directory specified by the **-P** argument.

Debugging NEMServer Configuration Errors

You can monitor your NEMServer communications by using any of several *trace* options, which are turned on and off in the *IOS.common.debug.exp* file located in the Cisco IP Manager subdirectory *SRVRS/ps/bin/nem/scripts*. To turn a trace feature on, set its value to 1; to turn a feature off, reset its value to 0.

Trace options are:

- **log_user**—when on, responses from all login sessions initiated by NEMServer are sent to **stdout** (normally, the window which displays NEMServer output).
- **debug_enable**—when on, the arguments sent to the *expect* script by the server are displayed in **stdout**.
- **log_enable**—when on, a line-by-line trace of the *expect* script execution is sent to a file called *config.log* in the directory from which NEMServer was launched.
- **exp_internal**—when on, the line-by-line trace of the *expect* script execution is sent to **stdout**.

The *setTelnetTraceOn* and *setTelnetTraceOff* utilities located in the *SRVRS* subdirectory can be used to set the **log_user** option.

Detecting Banner Text Errors

If banner text contains the characters # and > (used by IOS as part of a prompt), download operations can fail when the console connect method is used (see “Element Properties” in the chapter “Managing Network Elements”).

To enable a check for illegal banners, open the *IOS.common.debug.exp* file located in the Cisco IP Manager subdirectory *SRVRS/ps/bin/nem/scripts* and set the *check_prompt* flag to 1. To disable the check, set the flag to 0 (the default).

The test which detects illegal banners imposes a four-second delay on each download operation. To change the length of the delay, set the *banner_delay* flag to the desired time, in seconds.

Note The default four-second delay is a compromise. It may not be sufficient if there is heavy network traffic or if the script must decipher a very long banner. It is strongly recommended that the characters # and > not be used in banner text.

You can use the *setCheckPromptOn* and *setCheckPromptOff* scripts described in the “Other Utilities” section of the “Getting Started” chapter to turn the check on or off, but you cannot reset the length of the delay with these utilities.

NEMServer Command line options

Required

```
-u <database user name>
-p <database password>
```

Optional

```
-A // launch with -A to use Authorization and Access Control
-f <TFTP server name> // if none specified, TFTP will not be used
-i <database minimum connections> (defaults to 1)
-L <Cisco IP Manager log output>

   Can be server, cout, console, or <filename>. If server, then output
   will be sent to the LOGServer for storage in the database. If -L is
   not specified, no log messages will be generated.

-n <database name>
-P <TFTP subdirectory>

   Use this flag only if you want to specify a subdirectory of the
   directory specified in inetd.conf as your TFTP directory (must be
   /tftpboot). If inetd.conf specifies /tftpboot as the TFTP directory
   and you want to use /tftpboot/myDirectory (directory must already
   exist), enter the following command line argument:

      -P myDirectory

   If you want to use the directory specified in inetd.conf, do not use
   the -P flag

-q <thread pool size> (defaults to 20)
-r <Telnet retry count> (range: 1..5; defaults to 3)
-s <remote database server name>

   With an Oracle database, both -n and -s flags are equivalent to
   $ORACLE_SID

-T <Telnet timeout interval> (range: 3..10; defaults to 5 seconds)
-V // launch server executable with the -V flag to see version number
-x <database maximum connections> (defaults to 5)
```

NS (Naming Server)

The Naming Server helps applications find initial CORBA server object references using server names. There is no user command-line access.

Server Launch Default Values

The default values for various server flags are the values that will be in effect if you do not specify that flag when the server is started.

These values should not be confused with the default values assigned when *ipmgr.launch* is run. When this script is used, it will set the server flags to values that are considered appropriate by Cisco as a starting point for your installation.

This is a default installation; it does not mean that the servers are started up with default (or unmodified) values.

Editing Server Flags

Launch flags for each of the servers are defined in the Cisco-provided file *ipmgr.launch.csh*. When this file is sourced by the script *ipmgr.launch*, the environment variables used to set the server command-line launch flags are defined as follows:

Table A-1 Server Launch Script Defaults

Variable	Flags
AAS_CL_ARGS	-u <database user id specified in ipmgr.cfg.csh> -p <database password specified in ipmgr.cfg.csh> -L server -s <Oracle SID specified in ipmgr.cfg.csh> -n <Oracle SID specified in ipmgr.cfg.csh>
AUTH_CL_ARGS	-u <database user id specified in ipmgr.cfg.csh> -p <database password specified in ipmgr.cfg.csh> -L server -s <Oracle SID specified in ipmgr.cfg.csh> -n <Oracle SID specified in ipmgr.cfg.csh>

CNGS_CL_ARGS	-serverName CNGSServer
	     // Change the serverName flag if launching
	     // a second CNGSServer instance. Do not change
	     // any of the other flags.
	-home $NSM_ROOT
	-dataroot $ECSP_DATA/.dataroot
	-b $ECSP_DATA/baseline1
	-persistent

Table A-1 Server Launch Script Defaults (Continued)

CTM_CL_ARGS	-u <database user id specified in ipmgr.cfg.csh> -p <database password specified in ipmgr.cfg.csh> -A -L server -s <Oracle SID specified in ipmgr.cfg.csh> -n <Oracle SID specified in ipmgr.cfg.csh>
ES_CL_ARGS	Domain_Creation Domain_Modification Domain_Deletion Permission_Creation Permission_Modification Permission_Deletion User_Creation User_Modification User_Deletion Usergroup_Creation Usergroup_Modification Usergroup_Deletion Device_Creation Device_Modification_Comm Device_Modification_Config Device_Modification_Admin Device_Deletion Template_Creation Template_Modification Template_Deletion Templatedata_Creation Templatedata_Modification Templatedata_Deletion -nonames
ING_CL_ARGS	“ “ // INGServer is launched without a command line // argument
LOG_CL_ARGS	-u <database user id specified in ipmgr.cfg.csh> -p <database password specified in ipmgr.cfg.csh> -A -L server -s <Oracle SID specified in ipmgr.cfg.csh> -n <Oracle SID specified in ipmgr.cfg.csh>
NAM_CL_ARGS	-u <database user id specified in ipmgr.cfg.csh> -p <database password specified in ipmgr.cfg.csh> -A -L server -s <Oracle SID specified in ipmgr.cfg.csh> -n <Oracle SID specified in ipmgr.cfg.csh>
NEM_CL_ARGS	-u <database user id> -p <database password> -A -L server -f <TFTP server IP address specified in ipmgr.cfg.csh> -s <Oracle SID specified in ipmgr.cfg.csh> -n <Oracle SID specified in ipmgr.cfg.csh>
NS_CL_ARGS	“ “ // NS should be launched without command-line // arguments

These environment variables are not retained in memory after the conclusion of the process in which *ipmgr.launch* is executed, so they cannot be reviewed by displaying your environment variables in *stdout*.

To change the values of the **-u, -p, -s** or **-n** flags for any of the servers, you should rerun *ipmgr.configure* and specify different values when asked for the Oracle Server ID (**-s** and **-n** flags), Oracle user ID (**-u**) and Oracle password (**-p**).

Note The **-s** and **-n** flags may be set to different values for compatibility with other databases, if the Cisco IP Manager software is enhanced in the future; for an Oracle database, they are both set to the same value—the ORACLE_SID value.

To change the values of the **-f** and **-P** flags for NEMServer, rerun *ipmgr.configure* and specify a different host name or IP address for the TFTP server (**-f**) and a directory name when asked for a TFTP directory (**-P**; must be relative to */tftpboot*).

To launch all servers without AAS control, open the file *ipmgr.launch.csh* in a text editor and search for the following line:

```
setenv IPMGR_AAS_FLGS "-A"
```

Remove the **-A** flag, so that the line reads as follows:

```
setenv IPMGR_AAS_FLGS ""
```

Alternatively, you can search for the environment variable that sets each server and remove $IPMGR_AAS_FLGS from each.

To change the **-L** flag for all servers, search for the following line:

```
setenv IPMGR_LOG_FLGS "-L server"
```

Change *server* to *cout, console*, or a filename, or set the entire flag to “ ”.

Alternatively, you can search for the environment variable that sets each server and remove $IPMGR_LOG_FLGS from each, or change it to *“-L <option>”* (include the quote marks) as desired for each server.

If you are using the Cisco IP Manager API to develop your own user interface, you can add the **-htmlDirectory** argument to the CNGSServer flag by searching for the following line:

```
setenv CNGS_CL_ARGS "-serverName CNGSServer -home $NSM_ROOT -dataroot
$ECSP_DATA/.dataroot -b $ECSP_DATA/baseline1 -persistent"
```

Add the **-htmlDirectory** argument and its value to the string enclosed in quotes. Change the **-serverName** value only when launching a secondary CNGSServer instance. Do not change any of the other values.