Sun Java™ System

Instant Messaging 7 Administration Guide

2005Q1

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819-0430-10

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at http://www.sun.com/patents and one or more additional patents or pending patent applications in the U.S. and in other countries.

THIS PRODUCT CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF SUN MICROSYSTEMS, INC. USE, DISCLOSURE OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSION OF SUN MICROSYSTEMS, INC.

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, the Duke logo, the Java Coffee Cup logo, the Solaris logo, the SunTone Certified logo and the Sun ONE logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc.

Legato and the Legato logo are registered trademarks, and Legato NetWorker, are trademarks or registered trademarks of Legato Systems, Inc. The Netscape Communications Corp logo is a trademark or registered trademark of Netscape Communications Corporation.

The OPEN LOOK and Sun(TM) Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this service manual are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Sun Microsystems, Inc. détient les droits de propriété intellectuels relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs des brevets américains listés à l'adresse http://www.sun.com/patents et un ou des brevets supplémentaires ou des applications de brevet en attente aux Etats - Unis et dans les autres pays.

CE PRODUIT CONTIENT DES INFORMATIONS CONFIDENTIELLES ET DES SECRETS COMMERCIAUX DE SUN MICROSYSTEMS, INC. SON UTILISATION, SA DIVULGATION ET SA REPRODUCTION SONT INTERDITES SANS L AUTORISATION EXPRESSE, ECRITE ET PREALABLE DE SUN MICROSYSTEMS, INC.

Cette distribution peut comprendre des composants développés par des tierces parties.

Des parties de ce produit peuvent être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, le logo Duke, le logo Java Coffee Cup, le logo Solaris, le logo SunTone Certified et le logo Sun[tm] ONE sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

Legato, le logo Legato, et Legato NetWorker sont des marques de fabrique ou des marques déposées de Legato Systems, Inc. Le logo Netscape Communications Corp est une marque de fabrique ou une marque déposée de Netscape Communications Corporation.

L'interface d'utilisation graphique OPEN LOOK et Sun(TM) a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une license non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de ce manuel d'entretien et les informations qu'il contient sont regis par la legislation americaine en matiere de controle des exportations et peuvent etre soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucleaires, des missiles, des armes biologiques et chimiques ou du nucleaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou reexportations vers des pays sous embargo des Etats-Unis, ou vers des entites figurant sur les listes d'exclusion d'exportation americaines, y compris, mais de maniere non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une facon directe ou indirecte, aux exportations des produits ou des services qui sont regi par la legislation americaine en matiere de controle des exportations et la liste de ressortissants specifiquement designes, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE “EN L'ETAT” ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

Contents

Preface

Instant Messaging enables end users to participate in real-time interactive messaging and discussions. Sun Java System Instant Messaging allows end users to participate in Instant Messaging and chat sessions, send alert messages to each other, and share group news instantly. It is suitable for both intranets and the Internet.

This preface contains the following sections:

- Who Should Use This Book
- Before You Read This Book
- How This Book Is Organized
- Conventions Used in This Book
- Related Documentation
- Accessing Sun Resources Online
- Contacting Sun Technical Support
- Related Third-Party Web Site References
- Sun Welcomes Your Comments

Who Should Use This Book

You should read this book if you are responsible for administering, configuring, and deploying Instant Messaging.

Before You Read This Book

This book assumes that you are responsible for configuring, administering, and maintaining Instant Messaging, and you have an understanding of JavaScript™, HTML, and any of the following servers in your deployment:

- Sun Java™ System Portal Server
- A web container such as Sun Java™ System Application Server SE (Standard Edition)
- An SMTP server such as Sun Java™ System Messaging Server
- An LDAP server such as Sun Java™ System Directory Server
- Sun Java™ System Calendar Server
- Sun Java™ System Access Manager

How This Book Is Organized

The first chapter of this book provides an overview of the entire Instant Messaging product. The following table summarizes the content of this book in three parts.

Table 1 How This Book Is Organized

Chapter	Description
Preface	(This chapter)
Part I, "Postinstallation Configuration"	
Chapter 1, "Configuring Instant Messaging After Installation"	Contains instructions on configuration steps you need to complete after you install or upgrade, before you can use Instant Messaging.
Chapter 2, "Setting up and Launching Instant Messenger"	Provides information about configuring client systems, enabling Java^{TM} Web Start, and adding additional localization client files. Also explains how to launch the client.
Part II, "Administering Instant Messaging"	
Chapter 3, "Configuration File and Directory Structure Overview"	This chapter provides information about the configuration files you use to administer Instant Messaging.
Chapter 4, "Administering Instant Messaging Components"	This chapter describes how to administer Sun Java System Instant Messaging server and multiplexor.

Table 1 How This Book Is Organized *(Continued)*

Chapter	Description
Chapter 5, “Managing Instant Messenger”	This chapter describes how to customize and administer the Sun Java System Instant Messenger.
Chapter 6, “Managing Instant Messaging and Presence Policies”	This chapter describes how to manage administrator and end user privileges, especially with policies set in the Sun Java System Access Manager.
Chapter 7, “Managing The Instant Messaging Archive”	This chapter explains how to manage and configure the Instant Messaging Archive.
Chapter 8, “Troubleshooting and Monitoring Instant Messaging”	This appendix lists the common problems that might occur during installation and deployment of the Sun Java System Instant Messaging server.
Part III, “Reference Information”	
Appendix A, “Instant Messaging Configuration Parameters”	This appendix describes the settings you can configure for Instant Messaging.
Appendix B, “Instant Messaging imadmin Tool Reference”	This appendix describes the imadmin command used to administer Instant Messaging.
Appendix C, “Instant Messaging APIs”	This appendix explains the APIs used by Instant Messaging.

Conventions Used in This Book

The tables in this section describe the conventions used in this book.

Typographic Conventions

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

Table 2 Typographic Conventions

Typeface	Meaning	Examples
`AaBbCc123` (Monospace)	API and language elements, HTML tags, web site URLs, command names, file names, directory path names, onscreen computer output, sample code.	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`

`AaBbCc123`
(Monospace bold)
	What you type, when contrasted with onscreen computer output.	`%` **`su`**
		`Password:`

AaBbCc123 (Italic)	Book titles, new terms, words to be emphasized. A placeholder in a command or path name to be replaced with a real name or value.	Read Chapter 6 in the *User's Guide*. These are called *class* options. Do *not* save the file. The file is located in the *install-dir*`/bin` directory.

Symbols

The following table describes the symbol conventions used in this book.

Table 3 Symbol Conventions

Symbol	Description	Example	Meaning
`[ ]`	Contains optional command options.	`ls [-l]`	The `-l` option is not required.
`{ | }`	Contains a set of choices for a required command option.	`-d {y|n}`	The `-d` option requires that you use either the `y` argument or the `n` argument.
-	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.
+	Joins consecutive multiple keystrokes.	Ctrl+A+N	Press the Control key, release it, and then press the subsequent keys.
>	Indicates menu item selection in a graphical user interface.	File > New > Templates	From the File menu, choose New. From the New submenu, choose Templates.

Shell Prompts

The following table describes the shell prompts used in this book.

Table 4 Shell Prompts

Shell	Prompt
C shell	*machine-name*%
C shell superuser	*machine-name*#
Bourne shell and Korn shell	`$`
Bourne shell and Korn shell superuser	`#`
Windows command line	`C:\`

Related Documentation

The http://docs.sun.com^{SM} web site enables you to access Sun technical documentation online. You can browse the archive or search for a specific book title or subject.

Java Enterprise System Documentation

- *Sun Java Enterprise System Installation Guide*

  http://docs.sun.com/doc/819-0056

- *Sun Java Enterprise System 2005Q1Upgrade and Migration Guide*

  http://docs.sun.com/doc/819-0062

- *Sun Java System Communications Services Deployment Planning Guide*

  http://docs.sun.com/doc/819-0063

Other Server Documentation

For other server documentation, go to the following:

http://docs.sun.com/app/docs/prod/entsys#hic

This includes documentation sets for the following products:

- Sun Java System Directory Server
- Sun Java System Messaging Server
- Sun Java System Calendar Server
- Sun Java System Instant Messaging
- Sun Java System Access Manager
- Sun Java System Portal Server
- Sun Java System Web Server

Accessing Sun Resources Online

For product downloads, professional services, patches and support, and additional developer information, go to the following:

- Download Center

  `http://wwws.sun.com/software/download/`

- Professional Services

  `http://www.sun.com/service/sunps/sunone/index.html`

- Sun Enterprise Services, Solaris Patches, and Support

  `http://sunsolve.sun.com/`

- Developer Information

  `http://developers.sun.com/prodtech/`

Contacting Sun Technical Support

If you have technical questions about this product that are not answered in the product documentation, go to `http://www.sun.com/service/contacting`.

Related Third-Party Web Site References

Third-party URLs are often referenced in Sun documentation to provide additional, related information.

Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources.

Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions.

To share your comments, go to `http://docs.sun.com` and click Send Comments. In the online form, provide the document title and part number. The part number is a seven-digit or nine-digit number that can be found on the title page of the book or at the top of the document. For example, the title of this book is *Sun Java System Instant Messaging 7 2005Q1 Administration Guide*, and the part number is 819-0430-10.

Part I

Postinstallation Configuration

Chapter 1, “Configuring Instant Messaging After Installation”

Chapter 2, “Setting up and Launching Instant Messenger”

Chapter 1

Configuring Instant Messaging After Installation

After installation, you need to complete a few configuration steps before using Sun Java™ System Instant Messaging. This chapter describes these initial configuration steps in the following sections:

- Completing the Configuration Checklist
- Creating a UNIX System User and Group
- Configuring Instant Messaging After Installing or Upgrading
- Performing a Silent Configuration

Before you configure Instant Messaging, you should read and understand the information in the *Sun Java System Communications Services Deployment Planning Guide*, perform the installation as described in *Sun Java Enterprise System Installation Guide*, complete the configuration checklist, and finally configure the software.

Completing the Configuration Checklist

You should gather this information before you begin. You will be prompted for some or all of the information depending on the components you installed.

Print out Table 1-1 and write the values for your deployment in the space provided. You can reuse this checklist for multiple installations of Instant Messaging. This table contains passwords and other sensitive information, so you should store this information in a safe place.

Table 1-1 Configuration Parameters for Instant Messaging

Parameter	Description
Installation Directory	*im_svr_base* Directory in which Instant Messaging is installed. By default, Instant Messaging is installed into the `/opt` directory as follows: Solaris: `/opt/SUNWiim` Linux: `/opt/sun/im`
Instant Messaging Server Host and Domain Name	Host name on which Instant Messaging is installed and the domain name associated with the host. For example: Host Name: `instantmessaging.siroe.com` Domain Name: `siroe.com`
Instant Messaging Server Port Number	The port number on which the Instant Messaging Server listens for incoming requests from the multiplexor. Default: 45222
Instant Messaging Server-to-Server Port Number	The port number on which the Instant Messaging server listens for incoming requests from other Instant Messaging servers. In addition, if no multiplexor is installed, the server listens for incoming requests from Instant Messenger clients on this port. Default: 5269
Multiplexor Port Number (Multiplexor Configuration Only)	The port number on which the Instant Messaging Server listens for incoming requests from Instant Messenger clients. Default: 5222
Instant Messaging SSL Port	The port used for secure server-to-server communications. Default: 5223
Disable Server	Select this option if the instance you installed will act as a multiplexor and not a server. If you select this option, you must provide a value for Remote Instant Messaging Server Host Name.
Remote Instant Messaging Server Host Name (Multiplexor Configuration Only)	The host name of the Instant Messaging Server for which this multiplexor routes messages. If the multiplexor and server are installed on the same host, use `localhost`. Dependencies: The Disable Server parameter must be selected, that is, server functionality is disabled.
Assign Instant Messaging Services to existing users (Optional)	If selected, this option enables Instant Messaging for existing Sun Java System Access Manager users. Dependencies: Sun Java System Portal Server and Sun Java System Access Manager.

Table 1-1 Configuration Parameters for Instant Messaging *(Continued)*

Parameter	Description
Enable Instant Messaging Archive (Optional)	If selected, enables Sun Java System Portal Server search-based archiving for Instant Messaging. Dependencies: Sun Java System Portal Server and Sun Java System Access Manager.
LDAP Host Name	In a deployment with an LDAP server, the host name of the LDAP server that contains user and group information for Instant Messaging. For example, `directory.siroe.com`. Dependencies: LDAP server such as Sun Java System Directory Server.
LDAP Port Number	In a deployment with an LDAP server, the port number on which the directory server listens for incoming requests. For example, `389`. Dependencies: LDAP server such as Sun Java System Directory Server.
Base DN	In a deployment with an LDAP server, the base distinguished name in the directory tree that contains user and group information for Instant Messaging. For example, `o=airius.com`. Dependencies: LDAP server such as Sun Java System Directory Server.
Bind DN	In a deployment with Sun Java™ System Access Manager, during installation, you must use the Directory Manager Bind DN and password. The Bind DN is used to update the directory schema with the Instant Messaging and presence service templates and attributes only. This requires Directory Manager access. The Directory Manager Bind DN and password are not saved or used beyond installation and initial configuration. In a deployment with an LDAP server, for server configuration, Instant Messaging uses this Bind DN to search users and groups in the directory. Leave this blank if the directory can be searched anonymously. Dependencies: LDAP server such as Sun Java System Directory Server.
Bind Password	In a deployment with an LDAP server, the Bind DN password.
SMTP Server Host Name (Optional)	The host name of the SMTP server used to send email notification of messages to offline users. For example, `mail.siroe.com`. If the SMTP server does not use port 25, specify the port along with the host name. For example, if the SMTP server uses port `1025`: `mail.siroe.com:1025` Dependencies: SMTP server such as Sun Java System Messaging Server.

Table 1-1 Configuration Parameters for Instant Messaging *(Continued)*

Parameter	Description
Database, Logs, and Runtime File Pathname	The location where the runtime files, database, and logs are stored. Also referred to as *im_runtime_base*. Defaults: Solaris: `/var/opt/SUNWiim/default` Linux: `/var/opt/sun/im/` In addition, the database directory is often referred to as *im_db_base*. The defaults are as follows: Solaris: `/var/opt/SUNWiim/default/db` Linux: `/var/opt/sun/im/db/`
Resources and Help Files Pathname	Resource Directory. The directory in which the resource and online help files are installed. If you want to customize the resource files for your deployment, you should run `configure` utility, customize the files, then redeploy the resource files. You need to run configure first because the `configure` utility creates some of the index and `.jnlp` files that you can customize. See "Redeploying Resource Files" on page 101 for information. Default: *im_svr_base*`/html`

Table 1-1 Configuration Parameters for Instant Messaging *(Continued)*

Parameter	Description
Codebase	The URL from which Instant Messenger accesses resources, including the start page for initial downloads of the Instant Messaging client. The installation program installs the resource files into the following locations: Linux: `/opt/sun/im/html` Solaris: `/opt/SUNWiim/html` The `configure` utility uses the codebase to determine which web container instance to use. If it succeeds, the `configure` utility deploys the Instant Messenger resources as a web application in the web container, according to the URL provided. If no supported web container is detected, you will be prompted for a file system location in which to copy or link the resources. If you are using Instant Messaging with Sun Java™ System Application Server or Sun Java™ System Web Server, the `configure` utility automatically publishes the resource files to the web container for you. For Sun Java System Application Server, the `configure` utility uses the autodeploy mechanism, and for Sun Java System Web Server, the `configure` utility uses the `wdeploy` command. If you are using a different web container, the `configure` utility copies the files to a location you specify. This should include the web container's doc root. Alternatively, you can add the resource files installation directory as a doc root in your web container's configuration. See the documentation for your web container for more specific instructions. In addition, you can use a symbolic link to make the resources visible to the web container. For example, on Solaris the resources can be made visible to the web container by creating the following symbolic link: `ln -s /opt/SUNWiim/html` *docroot*`/im` Where *docroot* is the doc root of the web container, for example `/opt/web`. If you are using SSO with Sun Java™ System Access Manager, the Access Manager server and Instant Messaging server must be configured to use the same web container. See your web container documentation for more information about deploying resource files as a web application. See "Changing the Codebase" on page 76 if you need to modify the location of the resource files after initial configuration.

Creating a UNIX System User and Group

System users run specific server processes. Certain privileges need to be designated for these users to ensure they have appropriate permissions for the processes they run. Normally, the `configure` utility creates the following users and groups:

- User: `inetuser`
- Group: `inetgroup`

If the `config` utility does not create a UNIX user and group for Instant Messaging, you need to create them manually as described in this section. After you create the user and group for Instant Messaging, you should then set permissions appropriately for the directories and files owned by that user.

Do not choose `root` as a server user ID unless you are deploying Instant Messaging with Access Manager. In this case, you need to use `root` in order to allow access to the Access Manager configuration.

To create the appropriate user and group, follow these steps:

1. Log in as superuser.
2. Create a group to which your system user will belong. For example, to create a group named `imgroup` on Solaris, type the following:

   ```
   # groupadd imgroup
   ```

3. Create the system user and associate it with the group you just created. In addition, set the password for that user. For example, to create a user named `imuser` and associate it with the group `imgroup` on Solaris, type the following:

   ```
   # useradd -g imgroup imuser
   ```

   For more information on adding users and groups, refer to your operating system documentation.

4. Ensure that the user and group have been added to the `/etc/groups` file.

Configuring Instant Messaging After Installing or Upgrading

The Instant Messaging component is not configured by the Instant Messaging installer. Instead, you need to run the `configure` utility after you install the software.

If you want to customize the resource files for your deployment, you should run `configure` utility, customize the files, then redeploy the resource files. You need to run configure first because the `configure` utility creates some of the index and `.jnlp` files that you can customize. See "Redeploying Resource Files" on page 101 for information. Also see "Resources and Help Files Pathname" on page 22 for information on locating these files after configuration.

If you are using the BEA web container, you need to create a PASSFILE before you can configure Instant Messaging. If you are not using the BEA Web Container, skip to "To Configure Instant Messaging After Installation".

➤ To Create the PASSFILE for the BEA Web Container

1. Create a file named `installation directory/SUNWiim/lib/PASSFILE`.
2. Add the following lines to the file you created:

   `DS_DIRMGR_DN=`*Bind DN* for the Directory Manager
   `DS_DIRMGR_PASSWORD=`*Bind Password* for the Directory Manager
   `DS_HOST=`*LDAP Host Name*
   `DS_PORT=`*LDAP Port Number*
   `DS_BASE_DN=`*Base DN*

3. Fill in the values for each of the variables.

➤ To Configure Instant Messaging After Installation

1. Change to the directory in which you installed Instant Messaging.

   By default, this directory is `/opt/SUNWiim` on Solaris, and `/opt/sun/im` on Linux.

2. Run the `configure` utility in one of the following ways:

   Graphical user interface: **`configure`**

   Command-line: **`configure -nodisplay`**

   From a state file: **`configure -nodisplay -noconsole -state`** <*statefile*>

   where <*statefile*> is the path to the state file you want to use. If you are configuring using a state file, you will not be prompted for configuration information. Instead, the values from the state file are used to configure the software. See "Performing a Silent Configuration" on page 27 for information on generating a state file.

   If you are configuring using the graphical user interface or the command line, a series of prompts appears, requesting information that will set up the initial configuration for Instant Messaging. The prompts that appear vary depending on the components you installed. Fill in the requested information using the values from your Instant Messaging checklist. See "Completing the Configuration Checklist" on page 19 for information.

3. If you install the Sun Java System Access Manager on a different host from the Instant Messaging server, you need to manually copy the `imServices` files from the Instant Messaging server host to the Access Manager host after you run the `configure` utility. To do this:

   a. Locate the `imService_*.properties` files on the Instant Messaging server host.

      By default, these files are located under `/opt/SUNWiim/lib/` on Solaris and `/opt/sun/im/lib/` on Linux.

   b. Copy the files to the `locale` directory on the Sun Java System Access Manager host.

      By default this directory is `/opt/SUNWam/locale` on Solaris and `/opt/sun/identity/locale` on Linux.

4. After running the configuration utility, you need to configure the web container and client systems to support Instant Messaging. For instructions, see Chapter 2, "Setting up and Launching Instant Messenger".

Performing a Silent Configuration

To run a silent configuration, you first complete a false configuration to create a *state file*. During this `configure` session, your responses to the `configure` utility are captured in the state file, but no software is modified. In the state file, your responses are retained as a list of parameters, each representing a single prompt or field. Next, you will create a platform-appropriate state file ID and modify the state file to include this ID.

You can then run the `configure` utility on many hosts using the state file as input. This process allows you to quickly propagate one configuration across multiple hosts in your enterprise. See "Configuring Instant Messaging After Installing or Upgrading" on page 25 for information on using the state file to configure a new instance of Instant Messaging.

➤ To Generate a Configure State File and ID for Instant Messaging

1. Log in as superuser.
2. Change to the directory in which you installed Instant Messaging.

   By default, this directory is `/opt/SUNWiim` on Solaris, and `/opt/sun/im` on Linux.
3. Run the `configure` utility by typing the following at the command-line:

   `configure [-nodisplay] -saveState` *<statefile>*

   Where *<statefile>* is the name you want to use for the state file.

   To use the state file to configure a different installation of Instant Messaging, use the following command:

   `configure -nodisplay -noconsole -state` *<statefile>*

   As you proceed through the `configure` utility, your answers are captured in the state file. When you complete the configuration, the state file is available in the location that you specified.

4. Generate a platform-appropriate state file ID by running the `configure` utility again, but this time with the `-id` option as follows:

   ```
   configure -id
   ```

   The command generates an encrypted identifier.

5. Copy the identifier and paste the value into the state file as the value for the `STATE_BEGIN` and `STATE_DONE` parameters.

For information on using the state file to configure a different installation of Instant Messaging, see "Configuring Instant Messaging After Installing or Upgrading" on page 25.

Chapter 2

Setting up and Launching Instant Messenger

This chapter contains information about configuring the web container and client systems to support Instant Messenger in the following sections:

- Enabling Java™ Web Start
- Configuring Client Systems
- Launching Instant Messenger

Enabling Java™ Web Start

To use Instant Messenger with Java Web start, you need to install the software, then configure your web container to work with Java Web Start. For instructions on installing Java Web Start, go to the following location:

`http://java.sun.com/products/javawebstart`

To enable Java Web Start support in your web container, you need to edit the web container's `mime.types` file to include the following definition for JNLP:

Content Type: `application/x-java-jnlp-file`

Suffix: `jnlp`

On Sun Java System Web Server Enterprise Edition

➤ To Add the MIME Type to Sun Java System Web Server

1. Type the following URL to access the administration server in your browser:

   `http://`*hostname.domain-name:administration_port*

   For example: `http://budgie.siroe.com:8888`

   Sun Java System Web Server displays a window prompting you for a user name and password.

2. Type the administration user name and password you specified during the web container installation.

   The web container displays the Administration Server page.

3. On the Manage Servers page, click Manage.

   The web container displays the Server Manager page.

4. Click the MIME Types link.

5. From the MIME file drop-down list, choose a MIME type to edit and click OK.

6. In the Global MIME Types page, select `type` from the Category drop-down list.

7. In the Content-Type text box, type:

   ```
   application/x-java-jnlp-file
   ```

8. In the File-Suffix text box, type:

   ```
   jnlp
   ```

9. Click New Type to create the MIME type.

10. Restart the web container for this change to take effect.

On Apache Web Container

➤ To Add the MIME Type to Apache Web Container

- Add the following line to the `mime.types` file:

  ```
  application/x-java-jnlp-file jnlp
  ```

  By default, this file is located in the Apache Web Container configuration directory.

Configuring Client Systems

If the client machine has the appropriate version of Java installed, there are no additional requirements to use either Java Plug-in or Java Web Start. Netscape Navigator v7 as well as the recent versions of the Mozilla browser include the latest version of Java, while Internet Explorer does not. See the Sun Java System Instant Messaging Release Notes for version requirements.

If the client machine does not have the required version of Java installed, you need to install Java Web Start. You can download and Install Java from the following location:

`http://www.java.sun.com/j2se`

You can download and install Java Web Start from the following location:

`http://www.java.sun.com/products/javawebstart`

Launching Instant Messenger

You can run Instant Messenger as an applet within a web browser, or as a standalone application as described in the following sections:

- Running Instant Messenger From a Web Browser
- Running Instant Messenger as a Standalone Application

Running Instant Messenger From a Web Browser

➤ **To Run Instant Messenger as an Applet Within a Web Browser:**

1. Start the web browser.

   For information on supported browsers, see the Sun Java System Instant Messaging Release Notes.

2. Go to the Instant Messaging home page. By default, the home page is stored as `index.html`. Use the following format to locate the Instant Messaging home page:

   `http://`*codebase*`/index.html`

   Where *codebase* is the URL that corresponds to the location of the resource files on the web container.

3. Click Use Java Plug-In.

   If you customized the home page and changed the link text, click the link that corresponds to running Instant Messenger as an applet within a browser. The link points to either `im.jnlp` (standard) or `imssl.jnlp` (secure mode).

   When the Instant Messenger session is established using the Java Plug-in, the browser window must be dedicated to its use.

   You cannot locate any other URLs with this browser window, nor can you close the browser window without terminating the Instant Messenger session.

Running Instant Messenger as a Standalone Application

➤ **To Run Instant Messenger as a Standalone Application:**

1. Start the web browser.

   For information on supported browsers, see the Sun Java System Instant Messaging Release Notes.

2. Go to the Instant Messaging home page. By default, the home page is stored as `index.html`. Use the following format to locate the Instant Messaging home page:

   `http://`*codebase*`/index.html`

   Where *codebase* is the URL that corresponds to the location of the resource files on the web container.

3. Click Start.

   If you customized the home page and changed the link text, click the link that corresponds to running Instant Messenger using Java™ Web Start. The link points to either `im.html` (standard) or `imssl.html` (secure mode).

   See "Customizing Instant Messenger" on page 77 for information on customizing the resource pages.

Part II

Administering Instant Messaging

Chapter 3

Configuration File and Directory Structure Overview

This chapter provides information about the configuration files you use to administer Instant Messaging. Familiarize yourself with the locations of these files before making changes to your deployment's configuration.

This section describes the Instant Messaging server directory structure and the properties files used to store Instant Messaging operational data and configuration information.

Instant Messaging Server Directory Structure

Table 3-1 shows the platform-specific directory structure for the Instant Messaging server.

Table 3-1 Instant Messaging server directories

Description	Solaris Location	Linux Location
Program Files These files include the native executable files, the library files in the `bin` or `lib` directory, the shell scripts in the `sbin` directory, the java classes, and templates files in the `lib` directory.	Instant Messaging Installation Directory (*im_svr_base*) The default value for the Installation Directory is: `/opt/SUNWiim`	Instant Messaging Installation Directory (*im_svr_base*) The default value for the Installation Directory is: `/opt/sun/im`

Table 3-1 Instant Messaging server directories *(Continued)*

Description	Solaris Location	Linux Location
Server Configuration files These files are in the Configuration Directory and include the `iim.conf` file and a subdirectory which contains all the server-wide access control files.	Instant Messaging Configuration Directory (*im_cfg_base*) The default value for the Configuration Directory is: `/etc/opt/SUNWiim/default/config` For convenience, the installer creates a symbolic link from `/etc/opt/SUNWiim/default/config` to `/opt/SUNWiim/config`.	Instant Messaging Configuration Directory (*im_cfg_base*) The default value for the Configuration Directory is: `/etc/opt/sun/im/default/config` For convenience, the installer creates a symbolic link from `/etc/opt/sun/im/default/config` to `/opt/sun/im/config`.
Runtime Directory Contains Instant Messaging Server Data. These files include the configurable directory for the files generated by the server at runtime. It includes the end user data in the database directory. It also contains the server, multiplexor, Calendar agent, and XMPP service log files, in the `log` directory.	Instant Messaging Runtime Directory (*im_runtime_base*) The default value for the Runtime Directory is: `/var/opt/SUNWiim/default`	Instant Messaging Runtime Directory (*im_runtime_base*) The default value for the Runtime Directory is: `/var/opt/sun/im`
Database Contains end user information such as the user and news channels directory.	Instant Messaging Database Directory (*im_db_base*) The default value for the Database Directory is: `/var/opt/SUNWiim/default/db`	Instant Messaging Database Directory (*im_db_base*) The default value for the Database Directory is: `/var/opt/sun/im/db`
Instant Messenger resources. These files contain HTML documents and `jar` files used by Instant Messenger. The top-most directory contains the locale-independent resources, and the locale-specific directories contain the localized resources.	Instant Messaging Resource directory (*im_svr_base*`/html`) The default value for the Resource Directory is: `/opt/SUNWiim/html`	Instant Messaging Resource directory (*im_svr_base*`/html`) The default value for the Resource Directory is: `/opt/sun/im/html`

Instant Messaging Server Configuration File

Instant Messaging stores all configuration options in the `iim.conf` file. For more information on the parameters and their values stored in this file, see "Instant Messaging Configuration Parameters" on page 161.

Instant Messaging Data

Instant Messaging server stores the following data used by Instant Messenger in the database directory (*im_db_base*), and is indicated by the `iim.instancevardir` parameter in the `iim.conf` file:

- End user properties, such as contact lists, messenger settings, subscribed news channels and access control (alternatively, these properties can be stored in LDAP).
- News channel messages and access rules.
- Alert Messages that are to be delivered. These messages are delivered and removed when the recipient logs in.
- Public conferences. This does not involve instant messages which are not persistent, but only properties of the conference objects themselves, such as access rules.

Chapter 4

Administering Instant Messaging Components

This chapter explains how to administer the Instant Messaging components (server, multiplexor, Calendar agent, and watchdog) and perform other administrative tasks, such as changing configuration parameters and managing logging.

This chapter contains the following sections, which describe the various administrative tasks in Instant Messaging:

- Administering End Users
- Registering New Users
- Stopping, Starting, and Refreshing Instant Messaging Components
- Changing Instant Messaging Server and Multiplexor Configuration Parameters
- Managing Logging
- Federating Deployment of Multiple Instant Messaging Servers
- Using SSL in Instant Messaging
- Managing Instant Messaging's LDAP Access Configuration
- Backing Up Instant Messaging Data

Administering End Users

The administrative tasks in Instant Messaging are listed in the preceding section and are described throughout the rest of this chapter. Take note of the methods—as explained subsequently—for provisioning and managing end users.

Instant Messaging does not provide bulk user provisioning tools. You need to use a directory bulk provisioning tool for provisioning multiple Instant Messaging end users. By default, Instant Messaging does not provide specific commands to add, modify, or delete Instant Messaging end users. However, you can customize Instant Messenger to allow users to add themselves to the directory. See "Registering New Users" on page 43 for information.

Likewise in an LDAP-only deployment, you cannot prevent an end user from using Instant Messenger. In an LDAP-only deployment, the only way to prevent end users from using Instant Messaging is to delete them from the directory. In a deployment using Sun Java System Access Manager policy attributes, you can prevent an end user from accessing Instant Messenger. If you deploy Instant Messaging with Access Manager, you should use the provisioning tools provided with Access Manager instead of allowing users to register themselves.

The administrator can manage Instant Messaging end users, using the Instant Messaging Administrator Access Control mechanism. For more information on Instant Messaging Administrator Access Control, see "Overview of Privacy, Security, and Site Policies" on page 104. If you are using Sun Java System Access Manager, then the Access Manager is used for provisioning Instant Messaging end users. For more information, see *Sun Java System Communications Services Deployment Planning Guide*.

CAUTION If you deny end users the privilege to set up watches on other end users by editing the `sysWatch.acl` file, the Instant Messenger's Main window is not displayed for these end users. This effectively denies end users the ability to send instant messages. However, end users would still be able to see alerts and news channels.

Registering New Users

You can customize Instant Messenger to allow new user registration. When a user registers, the Instant Messaging server uses the information provided during registration to perform an `ldapadd` operation to create a user entry in the directory.

NOTE If you are using Instant Messaging with Sun Java System Access Manager, you should not allow users to register using this method. Instead, you should use the provisioning tools provided with Access Manager.

To allow new user registration, you need to add an argument to the `im.jnlp.template` and `im.html.template` files, run the `configure` utility, then (if necessary) redeploy the resource files.

This section describes:

- Customizing Instant Messenger to Allow New User Registration
- Registering New Users

See Chapter 5, "Managing Instant Messenger" for more information about customizing resource files.

Customizing Instant Messenger to Allow New User Registration

When you customize the resource files to allow new user registration, a new button appears on the Login dialog box. Users click this button to access the New User Registration dialog box where they can register. When a user registers, their information is added to the LDAP directory.

➤ To Customize Instant Messenger to Allow New User Registration

1. Open the `im.jnlp.template` file in a text editor.

   By default this file is stored in *im_svr_base*/`html`.

2. Search for the line:

   ```
   <application-desc main-class="com.iplanet.im.client.iIM">
   ```

3. Add the following argument to the end of the section:

   ```
   <argument>register=true</argument>
   ```

4. Save and close the `im.jnlp.template` file.
5. Repeat steps 1 through 4 for `im.html.template`.
6. Run the configure utility, selecting the "Messenger Resources" component only when prompted for which components you want to configure. See "Configuring Instant Messaging After Installing or Upgrading" on page 25 for instructions.
7. If you are using Sun Java System Access Manager or Sun Java System Web Server, redeploy the resource files as described in "Redeploying Resource Files" on page 101.
8. Launch Instant Messenger.

   The I am a New User button should appear on the Login dialog box.

Registering New Users

Once you have added the new user registration argument to the `im.jnlp` and `im.html` files and redeployed the resource files users can register themselves.

➤ To Register as a New User

1. In a web browser, go to the Instant Messaging home page.
2. Click Start or click Use Java Plug-in.

   The Login dialog box appears, displaying the I am a New User button.

3. Click I am a New User.

   The New User Registration dialog box appears.

4. Enter the information in the fields provided and click OK.

   The information is stored in the directory.

Stopping, Starting, and Refreshing Instant Messaging Components

The `imadmin` command enables you to:

- Start and stop all Instant Messaging components (server, multiplexor, watchdog, and Calendar agent).
- Start and stop an individual Instant Messaging component.
- Refresh all Instant Messaging component configurations.
- Refresh an individual Instant Messaging component.
- Check the status of the Instant Messaging components.

The `imadmin` command-line utility can be executed only by root or a user who has administration rights to the system(s) on which the Instant Messaging server and multiplexor are running. This end user is typically the identity that the server runs as, and is designated during installation:

- On Solaris - `inetuser`
- In a deployment with Sun Java System Access Manager, if the Sun Java System Portal Server and the Instant Messaging server are installed on the same host, the user is the one who is running the Access Manager, as `root`.

The `imadmin` command-line utility is located in the following directory:

im_svr_base/`sbin`

Starting the Instant Messaging server enables Instant Messenger to connect to it. Stopping the Instant Messaging server closes all connections and disconnects all Instant Messenger clients.

To Start Instant Messaging Components

You can start all the components together or a single component separately.

Use the `imadmin` command to start the Instant Messaging Server, multiplexor, Calendar agent, and watchdog depending on which components are enabled:

➤ To Start All Components

- Use the `imadmin` command to start all components.

  ```
  imadmin start
  ```

  If both server and multiplexor are enabled, this command first starts the Instant Messaging server, and then starts the multiplexor.

  If the watchdog is enabled (default), this command starts the watchdog, then the watchdog reads the configuration file and starts the Instant Messaging Server and/or multiplexor as necessary.

➤ To Start a Single Component

- Use the `imadmin` command with an argument that designates the component as follows:

  Server: `imadmin start server`

  Multiplexor: `imadmin start multiplexor`

  Calendar agent: `imadmin start agent-calendar`

  Watchdog: `imadmin start watchdog`

To Stop Instant Messaging Components

You can stop all the components together or a single component separately.

Use the `imadmin` command to stop the Instant Messaging Server, multiplexor, Calendar agent, and watchdog depending on which components are enabled:

➤ To Stop All Components

- Use the `imadmin` command to stop all components:

  ```
  imadmin stop
  ```

  If the watchdog is running, the `imadmin` utility brings the watchdog down first, and then stops the server and/or the multiplexor.

  This command stops the server, multiplexor, Calendar agent, and watchdog, terminates all end user connections, and disconnects any inbound and outbound servers configured.

➤ To Stop a Single Component

- Use the `imadmin` command with an argument that designates the component as follows:

  Server: `imadmin stop server`

  Multiplexor: `imadmin stop multiplexor`

  Calendar agent: `imadmin stop agent-calendar`

  Watchdog: `imadmin stop watchdog`

To Refresh Component Configuration

Use the `imadmin` command with the `refresh` parameter to stop and restart an individual Instant Messaging component and refresh that component’s configuration.

You can refresh all the components together or a single component separately.

Whenever you change a configuration parameter in the `iim.conf` file, make sure to refresh the configuration.

➤ To Refresh All Components

- Use the `imadmin` command to refresh all components:

  ```
  imadmin refresh
  ```

  This command stops the server, multiplexor, Calendar agent, and watchdog, terminates all end user connections, and disconnects any inbound and outbound servers configured.

  If the watchdog is running, the `imadmin` utility brings the watchdog down first, and then stops the server and/or the multiplexor. Then starts the watchdog which reads the configuration file and starts the Instant Messaging Server and/or multiplexor as necessary.

➤ To Refresh a Single Component

- Use the `imadmin` command with an argument that designates the component as follows:

  Server: `imadmin refresh server`

  Multiplexor: `imadmin refresh multiplexor`

  Calendar agent: `imadmin refresh agent-calendar`

  Watchdog: `imadmin refresh watchdog`

To Check the Status of Instant Messaging Components

You can check the status of all the components together or a single component separately using the `imadmin` command.

➤ To Check the Status of All Components

- Use the `imadmin` command to check the status.

  ```
  imadmin check
  ```

  This command returns the status of all enabled components.

➤ To Check the Status of a Single Component

- Use the `imadmin` command with an argument that designates the component as follows:

  Server: `imadmin check server`

  Multiplexor: `imadmin check multiplexor`

  Calendar agent: `imadmin checkagent-calendar`

  Watchdog: `imadmin check watchdog`

Changing Instant Messaging Server and Multiplexor Configuration Parameters

Instant Messaging stores configuration parameters in the `iim.conf` file. For a complete list of configuration parameters, see Instant Messaging Configuration Parameters.

To change configuration parameters, manually edit the configuration parameters and values in the `iim.conf` file, then refresh the Instant Messaging server configuration. If you change a multiplexor parameter, you only need to refresh the multiplexor using the following `imadmin` command:

`imadmin refresh` multiplexor

For a complete list of parameters and their values, see "Instant Messaging Configuration Parameters" on page 161.

➤ To Change Configuration Parameters

1. Change to the `config` directory. For example, on Solaris type:

   ```
   cd etc/opt/SUNWiim/default/config
   ```

2. Edit the `iim.conf` file. For example:

   ```
   vi iim.conf
   ```

3. Save your changes.

4. Refresh the configuration.

CAUTION If you change the multiplexor listen port (`iim_mux.listenport`) or the multiplexor host, update the `im.html` or the `im.jnlp` files accordingly. Failure to do so disables Instant Messenger from connecting to the server. For more information, see the section on Managing Instant Messenger.

Managing Logging

Instant Messaging creates log files that record events, related status of various software components, system errors, and other aspects of the server, multiplexor, Calendar agent, and watchdog. By examining the log files, you can monitor many aspects of the server's operation. In addition, you can collect logging data for Instant Messenger on demand. This section provides information about logging in the following topics:

- Logging Overview
- Logging Levels
- Administering Client Logging

Logging Overview

You can configure the level of logging for the Instant Messaging server, multiplexor, Calendar agent, and watchdog by specifying the parameters in the `iim.conf` file. For information on configuring the level of logging in the `iim.conf` file, see "Changing Instant Messaging Server and Multiplexor Configuration Parameters" on page 49.

The location of the log files are specified during Instant Messaging configuration. Typically, log files are stored in *im_runtime_base*/`log`. Where the defaults for *im_runtime_base* are as follows:

- Solaris:

  `/var/opt/SUNWiim/default`

- Linux:

  `/var/opt/sun/im`

As part of regular system maintenance, you need to periodically review and trim the log files from occupying more disk space. The server does not perform this action.

Table 4-1 provides the name of the log files and the configuration parameter in `iim.conf` used to set the logging level for each log file.

Table 4-1 Log File Names and Logging Level Configuration Parameters

Component	Log File Name	Logging Level Configuration Parameter
Server	`xmppd.log`	`iim.log.iim_server.severity`
Multiplexor	`mux.log`	`iim.log.iim_mux.severity`
Calendar agent	`agent-calendar.log`	`iim.log.agent-calendar.severity`
Watchdog	`iim_wd.log`	`iim.log.iim_wd.severity`

The configuration parameters can have the following values:

- `fatal`
- `error`
- `warning`
- `info`
- `debug`

In addition, logging configuration in deployments with Sun Java System Access Manager is determined by the `com.iplanet.services.debug.level` property. You set this property in the `AMConfig.properties` file on the Sun Java System Access Manager host. By default, this file is installed in the following location:

AM_svr_base`/lib/AMConfig.properties`

Where *AM_svr_base* is the directory in which you installed Access Manager.

This property can contain the following values:

- `message`
- `warning`
- `error`
- `off`

By default, the Sun Java System Portal Server desktop log file (desktop.debug) and archive log files (IMArchiveSearch.log and IMArchiveSubmit.log) are stored in the following locations:

- Solaris: /var/opt/SUNWam/debug
- Linux: /var/opt/sun/am/debug

Logging Levels

The level or priority of maintaining the error log defines how detailed, or verbose, the log should be. A higher priority level implies less details as only events of high priority (high severity) are recorded in the log file. In contrast a lower priority level implies greater details as more events are recorded in the log file.

You can set the logging level separately for the Instant Messaging server, multiplexor, watchdog, and the Calendar agent.

Table 4-2 describes the logging levels for the components. These logging levels are a subset of the levels defined by the Unix syslog facility.

Table 4-2 Logging Levels for Instant Messaging Components

Level	Description
FATAL	This priority level records minimum logging details in the log file. A log record is added to the log file whenever a severe problem or critical condition occurs. If a FATAL problem occurs, the application might stop functioning.
ERROR	A log record is added to the log file whenever a recoverable software error condition occurs or a network failure is detected. For example, when the server fails to connect to a client or to another server.
WARNING	A log record is added to the log file whenever a user error is detected. For example, when the server cannot understand the communication sent by the client.
INFO	A log record is added to the log file whenever a significant action takes place. For example, when an end user successfully logs in or logs out.
DEBUG	The tasks are recorded in the log file. This information is useful for debugging purposes only. Each event with individual steps within each process or task are written to the log file, to help the end user identify the problems while debugging the application.

When you select a particular logging level, events corresponding to that level and to all higher and less verbose levels are logged.

`INFO` is the default level for the server. `ERROR` is the default level for the multiplexor and watchdog log files.

NOTE If you specify `DEBUG` as the logging level, your log files will occupy more disk space. Monitor and trim your log files to prevent them from occupying more disk space.

To Set Log Levels

You set the log levels by modifying parameters within the `iim.conf` file. Table 4-1 on page 51 contains a list of the log files and the parameter that you need to set for each component.

For more information on changing parameters, see "Changing Instant Messaging Server and Multiplexor Configuration Parameters" on page 49. For more information on the watchdog, see "Managing the Watchdog Process" on page 157. For more information on the Calendar agent, see "Using Calendar Pop-up Reminders" on page 96.

Administering Client Logging

By default, client data is not logged. You may be asked to collect client data during a support call. In this situation, you will need to enable logging before you can view client log data.

To enable client logging you need to complete the following steps:

1. Enable logging in Java Web Start Application Manager or the Java Plug-In Control Panel on the client's host.

2. Add a debug parameter to the `im.jnlp` file.

3. Redeploy the Instant Messenger resources if necessary.

➤ To Enable Client Logging

1. Enable the logging feature in either the Java Web Start Application Manager or the Java Plug-In Control Panel as appropriate.

   If the client uses the Java plug-in with an earlier version of the JDK, run the Java Plug-In Control Panel. See the online help for the Java Plug-In Control Panel for instructions on enabling logging.

   If the client uses Java Web Start or uses the plug-in with JDK 5.0, run the Java Web Start Application Manager, then:

   a. Select File | Preferences.

      The Preferences dialog box appears.

   b. On the Advanced tab, select the Log Output checkbox and enter a Log File Name.

   c. Click OK.

2. Open `im.jnlp` in a text editor.

3. Search for the line:

   ```
   <application-desc main-class="com.iplanet.im.client.iIM">
   ```

4. Add the following argument to the end of the section:

   ```
   <argument>debug=true</argument>
   ```

5. Save and close the `im.jnlp` file.

6. If you are using Sun Java System Access Manager or Sun Java System Web Server, redeploy the resource files as described in "Redeploying Resource Files" on page 101.

7. Relaunch Instant Messenger.

Federating Deployment of Multiple Instant Messaging Servers

In an LDAP-only deployment, when you federate multiple Instant Messaging deployments you form a larger Instant Messaging community. End users from different servers can communicate with each other, use conference rooms on other domains, and subscribe to news channels on remote servers based on the access privileges.

In a deployment with Sun Java System Access Manager, a single Instant Messaging server can host multiple domains. You can designate a single domain as the default domain for a Instant Messaging server instance. End users in different domains hosted by the same server cannot interact with each other. When you federate multiple Instant Messaging deployments, end users in default domains can see the end users in default domains of other remote Instant Messaging servers.

For enabling communication between multiple Instant Messaging servers in your network, you need to configure your server to identify itself to the other Instant Messaging servers in the network. A Instant Messaging server identifies itself with its domain name, host and port number, serverID, and password.

In an LDAP-only deployment, the two servers should reside in different domains.

Within the server configuration, you can assign each Instant Messaging server a symbolic name, consisting of letters and digits, for example, `IMserver1`.

CAUTION It is recommended that the server-to-server communication is secured using TLS (SSL). This is required to prevent third party infringement of security when data is exchanged between two servers. This precaution is extremely desirable in the case where the link between the two servers uses the public internet. Follow the instructions outlined below to configure SSL between Instant Messaging servers.

To Configure Communication Between Instant Messaging Servers

This procedure describes how to enable communication between two Instant Messaging servers, `iim.company22.com` and `iim.i-zed.com`..

➤ To Configure Communication Between Two Servers

1. Gather the following information listed in Table 4-3.

   Table 4-3 lists the parameters in the iim.conf file for server-to-server communication and the values for these parameters in the Instant Messaging servers, iim.company22.com and iim.i-zed.com.

Table 4-3 Configuration Information for Server-to-Server Communication

Parameter in iim.conf File	Value for Server iim.company22.com	Value for Server iim.i-zed.com
iim_server.serverid	Iamcompany22	Iami-zed
iim_server.password	secretforcompany22	secret4i-zed
iim_server.coservers	coserver1	coserver1
iim_server.domain	iim.company22.com	iim.i-zed.com
iim_server.coserver1.host	iim.i-zed.com:9919	iim.company22.com:9919
iim_server.coserver1.serverid	Iami-zed	Iamcompany22
iim_server.coserver1.password	secret4i-zed	secretforcompany22
iim_server.coserver1.domain	i-zed.com	company22.com

For more information on the configuration parameters, see Instant Messaging Configuration Parameters.

NOTE You can configure your server to communicate with other Instant Messaging servers. Each Instant Messaging server is identified by its symbolic name. The symbolic name of the server is added in the iim_server.coservers parameter in the iim.conf file. This parameter has multiple values and each value is separated by a comma.

2. Change to the config directory on the server iim.company22.com. For example, on Solaris:

   ```
   cd /etc/opt/SUNWiim/default/config
   ```

3. Edit the iim.conf file, for example:

   ```
   vi iim.conf
   ```

NOTE The iim.conf file should be owned by the Instant Messaging server account you created during installation. If the iim.conf file cannot be read by the Instant Messaging server account, Instant Messaging server and multiplexor would be unable to read the configuration. Additionally, you might lose the ability to edit the iim.conf file.

The following example shows the section of the iim.conf file on iim.company22.com corresponding to the server-to-server communications that you can modify:

```
iim_server.serverid=Iamcompany22
iim_server.password=secretforcompany22
iim_server.domain=iim.icompany22.com
iim_server.coservers=coserver1
iim_server.coserver1.host=iim.i-zed.com:9919
iim_server.coserver1.serverid=Iami-zed
iim_server.coserver1.password=secret4i-zed
iim_server.coserver1.domain=i-zed.com
```

4. Follow Step 2 through Step 3 for the iim.conf file on server iim.i-zed.com.

   The following example shows the section of the iim.conf file on iim.i-zed.com corresponding to the server-to-server communications that you can modify:

```
iim_server.serverid=Iami-zed
iim_server.password=secret4i-zed
iim_server.domain=iim.i-zed.com
iim_server.coservers=coserver1
iim_server.coserver1.host=iim.company22.com:9919
iim_server.coserver1.serverid=Iamcompany22
iim_server.coserver1.password=secretforcompany22
iim_server.coserver1.domain=company22.com
```

5. Save the changes and refresh the configurations on both servers.

Using SSL in Instant Messaging

Instant Messaging supports the Secure Sockets Layer (SSL) protocol, for encrypted communications and for certificate-based authentication of Instant Messaging servers. Instant Messaging server supports SSL version 3.0.

Instant Messaging multiplexor and Instant Messenger also support SSL for encrypted communication between the client and the multiplexor.

Enabling SSL for Instant Messaging server requires the following:

1. Obtaining and installing a certificate for your Instant Messaging server, and configuring the Instant Messaging server to trust the Certification Authority's certificate.

2. Ensuring that each Instant Messaging server that needs to communicate using SSL with your server, obtains and installs a certificate.

3. Turning on SSL in the server by setting the appropriate parameters in the `iim.conf` file.

Enabling SSL between the multiplexor and Instant Messenger requires the following:

1. Requesting a Certificate from the Certificate Authority
2. Installing the Certificate
3. Enabling SSL Between the Multiplexor and Instant Messenger
4. To Activate SSL for Server to Server Communication
5. Invoking the Secure Version of Instant Messenger

For more information about managing certificates, see the Web Server and Application Server product documentation at `http://docs.sun.com`

Requesting a Certificate from the Certificate Authority

To enable SSL between Instant Messenger and multiplexor, you need to install the certificate and create databases for secure communication. You can request and install the certificate using Web Server or Application Server.

➤ To Request and Install a Certificate

1. Type the following URL for starting the administration server in your browser:

   ```
   http://hostname.domain-name:administration_port
   ```

   A window prompting you for a user name and password appears.

2. Type the administration user name and password you specified during the Web Server or Application Server installation.

   The Administration Server page appears.

3. Create a separate Web Server or Application Server instance. For more information on installing multiple instances of the server, see the product documentation at:

   `http://docs.sun.com/`

4. Create a trust database to store the public and private keys, referred as the key-pair file. The key-pair file is used for SSL encryption.

   For information on creating a trust database, see the Web Server or Application Server product documentation at:

   `http://docs.sun.com/`

5. Request a certificate from the Certificate Authority.

   For more information on requesting a certificate, see the Web Server or Application Server product documentation at:

   `http://docs.sun.com/`

Installing the Certificate

When you receive the server certificate from your Certificate Authority, you need to install the certificate.

➤ To Install the Certificate

1. Type the following URL for starting the administration server in your browser:

   ```
   http://hostname.domain-name:administration_port
   ```

   A window appears, prompting you for a user name and password.

2. Type the administration user name and password you specified during the Web Server or Application Server installation.

   The Administration Server page appears.

3. Install the server certificate.

   For more information on installing the certificate, see the Web Server or Application Server product documentation at:

   http://docs.sun.com/

4. Change to your Web Server or Application Server's `alias` directory.

5. Copy the database files from the `alias` directory to the Instant Messenger `config` directory.

   To copy the database files from the `alias` directory to the Instant Messenger `config` directory, type the following:

   ```
   cp https-serverid-hostname-cert8.db
   /etc/opt/SUNWiim/default/config/cert8.db
   ```

   ```
   cp https-serverid-hostname-key3.db
   /etc/opt/SUNWiim/default/config/key3.db
   ```

   ```
   cp secmod.db /etc/opt/SUNWiim/default/config/secmod.db
   ```

NOTE The end user on which the Instant Messaging server runs should have Read permission on `cert7.db, key3.db,`and `secmod.db` files.

6. Change to your Instant Messaging *im_cfg_base* directory. For example, on Solaris:

   ```
   cd /etc/opt/SUNWiim/default/config
   ```

7. Create the `sslpassword.conf` file using an editor of your choice. For example, you could type:

   ```
   vi sslpassword.conf
   ```

8. Enter the following line to the `sslpassword.conf` file

   ```
   Internal (Software) Token:password
   ```

   Password: The password specified during the creation of the trust database.

9. Save the file.

NOTE All Instant Messenger end users should have Ownership and Read permission on the `sslpassword.conf` file.

10. After verifying the functioning of SSL, log in to the Web Server or Application Server as an administrator and remove the server instance that you have created while requesting the certificate.

Enabling SSL Between the Multiplexor and Instant Messenger

Table 4-4 lists the parameters in the `iim.conf` file for enabling SSL between Instant Messenger and multiplexor. It also contains the description and the default value of these parameters:

Table 4-4 Instant Messenger and Multiplexor SSL Parameters

Parameter	Default Value	Description
`iim_mux.usessl`	`off`	If the value is set to "`on`", the multiplexor requires an SSL handshake for each connection it accepts, before exchanging any application data.

`iim_mux.secconfigdir`	Solaris:
	`/etc/opt/SUNWiim/default/config`
	Linux:
	`/etc/opt/sun/im/default/config`
		This directory contains the key and certificate databases. It usually contains the security module database.

`iim_mux.keydbprefix`	(Empty string)	This value should contain the key database filename prefix. The key database file name must always end with `key3.db.` If the Key database contains a prefix, for example `This-Database-key3.db`, then value of this parameter is `This-Database`.
`iim_mux.certdbprefix`	(Empty string)	This value should contain the certificate database filename prefix. The certificate database file name must always end with `cert7.db`. If the certificate database contains a prefix, for example `Secret-stuff-cert7.db`, then value of this parameter is `Secret-stuff`.
`iim_mux.secmodfile`	`secmod.db`	This value should contain the name of the security module file.

Table 4-4 Instant Messenger and Multiplexor SSL Parameters *(Continued)*

Parameter	Default Value	Description
`iim_mux.certnickname`	`Server-Cert`	This value should contain the name of the certificate you entered while installing the certificate. The certificate name is case-sensitive.
`iim_mux.keystorepasswordfile`	`sslpassword.conf`	This value should contain the relative path and the name of the file containing the password for the key database. This file should contain the following line: `Internal (Software) Token:`*password* Where *password* is the password protecting the key database.

➤ To Enable SSL Between Instant Messenger and Multiplexor

1. Change to the *im_cfg_base* directory. For example, on Solaris:

```
cd /etc/opt/SUNWiim/default/config
```

2. Edit the `iim.conf` file, for example:

```
vi iim.conf
```

3. Add the values mentioned in the Table 4-4 to the multiplexor configuration parameters.

   The following is an example of the `iim.conf` file with the multiplexor configuration parameters:

```
! IIM multiplexor configuration
! ============================
!
! Multiplexor specific options

! IP address and listening port for the multiplexor.
! WARNING: If this value is changed, the port value of ’-server’ argument
! in the client’s im.html and im.jnlp files should also be changed to match
th
is.
iim_mux.listenport = "siroe.com:5222"

! The IM server and port the multiplexor talks to.
iim_mux.serverport = "siroe.com:45222"

! Number of instances of the multiplexor.
iim_mux.numinstances = "1"

! Maximum number of threads per instance
iim_mux.maxthreads = "10"

! Maximum number of concurrent connections per multiplexor process
iim_mux.maxsessions = "1000"

iim_mux.usessl = "on"
iim_mux.secconfigdir = "/etc/opt/SUNWiim/default/config"
iim_mux.keydbprefix = "This-Database"
iim_mux.certdbprefix = "Secret-stuff"
iim_mux.secmodfile = "secmod.db"
iim_mux.certnickname = "Server_Cert"
iim_mux.keystorepasswordfile = "sslpassword.conf"
```

Invoking the Secure Version of Instant Messenger

The secure version of Instant Messenger can be invoked by accessing the `imssl.html` file or `imssl.jnlp` file from your browser. These files are located under the resource directory, the base directory under which all the Instant Messenger resources are stored.

The links to these applet descriptor files can also be added to `index.html` file.

To Activate SSL for Server to Server Communication

Before you can activate SSL, you must create a certificate database, obtain and install a server certificate, and trust the CA's certificate as described earlier.

➤ To Activate SSL

1. Set these `iim.conf` parameters:

   - `iim_server.usesslport=true`
   - `iim_server.sslport=5223`

   These parameters should already be in the `iim.conf` file.

2. Set the server-to-server configurations as described in Federating Deployment of Multiple Instant Messaging Servers, and add the following:

   - `iim_server.coserver1.usessl=true`

   Change the port number of the following:

   - `iim_server.coserver1.host=hostname:5223`

   The port number should be the SSL port of the other server.

Following is a section of `iim.conf` file with the required SSL configuration:

```
! Server to server communication port.
iim_server.port = "5269”
! Should the server listen on the server to server communication port
iim_server.useport = "True”
! Should this server listen for server-to-server communication using ssl port
iim_server.usesslport = "True”
iim_server.sslport=5223
iim_server.coservers=coserver1
iim_server.coserver1.serverid=Iamcompany22
iim_server.coserver1.password=secretforcompany22
iim_server.coserver1.usessl=true
iim_server.coserver1.host=iim.i-zed.com:5223
iim_server.serverid=Iami-zed
iim_server.password=secret4i-zed
iim_server.secconfigdir = "/etc/opt/SUNWiim/default/config"
iim_server.keydbprefix = "This-Database"
iim_server.certdbprefix = "Secret-stuff"
iim_server.secmodfile = "secmod.db"
iim_server.certnickname = "Server_Cert"
iim_server.keystorepasswordfile = "sslpassword.conf"
```

Enabling SSL between two servers

Table 4-5 on page 65 lists the parameters in the `iim.conf` file for enabling SSL between two Instant Messaging servers. It also contains the description and the default value of these parameters:

Table 4-5 Server-to-Server SSL Configuration Parameters

Parameter	Default Value	Description
`iim_server.secconfigdir`	Solaris: `/etc/opt/SUNWiim/default/config` Linux: `/etc/opt/sun/im/default/config`	This directory contains the key and certificate databases. It usually contains the security module database.

Table 4-5 Server-to-Server SSL Configuration Parameters *(Continued)*

Parameter	Default Value	Description
`iim_server.keydbprefix`	None	This value should contain the key database filename prefix. The key database file name must always end with `key3.db.` If the Key database contains a prefix, for example `This-Database-key3.db`, then value of this parameter is `This-Database`.
`iim_server.certdbprefix`	None	This value should contain the certificate database filename prefix. The certificate database file name must always end with `cert7.db`. If the certificate database contains a prefix, for example `Secret-stuff-cert7.db`, then value of this parameter is `Secret-stuff`.
`iim_server.secmodfile`	`secmod.db`	This value should contain the name of the security module file.
`iim_server.certnickname`	`Server-Cert`	This value should contain the name of the certificate you entered while installing the certificate. The certificate name is case-sensitive.
`iim_server.keystorepasswordfile`	`sslpassword.conf`	This value should contain the relative path and the name of the file containing the password for the key database. This file should contain the following line: `Internal (Software) Token:`*password* Where *password* is the password protecting the key database.
`iim_server.trust_all_cert`	false	If this value is true than the server will trust all certificates and will also add the certificate information into the log files.

Managing Instant Messaging's LDAP Access Configuration

An LDAP-only deployment of Instant Messaging server requires a directory server. In an LDAP-only deployment, the Instant Messaging server uses the directory server to perform end-user authentication and to search for end users.

In a deployment with Sun Java System Access Manager, the Instant Messaging server uses the directory used by Sun Java System Portal Server. When installed in an Access Manager deployment environment, the Instant Messaging server uses the directory used by the Access Manager to search for end users, and not for end-user authentication. In an Access Manager deployment, Access Manager performs the authentication.

If you use an LDAP directory to maintain your user namespace, the default configuration makes the following assumptions regarding the schema used by this directory:

- End user entries are identified by the `inetOrgPerson` object class.
- Group entries are identified by the `groupOfUniqueNames` or `groupofURLs` object class.
- Instant Messenger user ID attribute of an end user is provided by the `uid` attribute (from `inetOrgPerson` objectclass).
- The email address of an end user is provided by the `mail` attribute.
- The display name of an end user or group is provided by the `cn` attribute.
- The list of members of a group is provided by the `uniqueMember` attribute (`groupOfUniqueNames` object class).

You can change these default settings by editing the `iim.conf` file. See "Using the iim.conf file" on page 161 for more information.

Searching the Directory as Anonymous Users

Instant Messaging needs to be able to search the directory to function correctly. If your directory is configured to be searchable by anonymous users, Instant Messaging has the capability to search the directory. If the directory is not readable or searchable by anonymous users, you must take additional steps to configure the `iim.conf` file with the credentials of a user ID that has at least read access to the directory.

These credentials consist of:

- A distinguished name (dn)
- The password of the above dn

➤ To Enable the Server to Conduct Directory Searches as a Specific End User

1. Identify values for the following parameters in the iim.conf file:

   - iim_ldap.usergroupbinddn - Specifies the distinguished name (dn) to use to bind to the directory for searches.
   - iim_ldap.usergroupbindcred - Specifies the password to use with the distinguished name (dn)

   For example:

   ```
   iim_ldap.usergroupbinddn="cn=iim server,o=i-zed.com"
   ```

   ```
   iim_ldap.usergroupbindcred=secret
   ```

NOTE You do not have to use administrator-level credentials with write level access, as all that is necessary is read access to the domain tree. Thus, if there is an LDAP user with read level access, use its credentials instead. This is a safer alternative as it does not force you to disseminate the administrator-level credentials.

2. In a deployment with Access Manager, the directory is generally not searchable by anonymous users. You should set the iim_ldap.useidentityadmin configuration parameter to true in this case. Also you can delete or comment out the following configuration parameters:

   - iim_ldap.usergroupbinddn
   - iim_ldap.usergroupbindcred.

3. Edit the iim.conf file.

   See "Changing Instant Messaging Server and Multiplexor Configuration Parameters" on page 49 for instructions on editing the iim.conf file.

   If the iim_ldap.usergroupbinddn and iim_ldap.usergroupbindcred parameters do not appear in the iim.conf file, you can add them anywhere in the file.

Configuring Dynamic LDAP Server Group

In the LDAP Server, the dynamic groups filter end users based on their DN and include them in a single group. The dynamic groups are defined in Directory Server as the `groupOfUrls` objectclass.

To enable end users to view the dynamic groups in search results and add them to their contact list, you need to include the `groupOfUrls` objects to search results.

The following modifications need to be made to the server configuration file `iim.conf`:

1. Change to the `config` directory. For example, on Solaris:

   `cd /etc/opt/SUNWiim/default/config`

2. Edit the `iim.conf` file. For example:

   `vi iim.conf`

3. Add the following information to the `iim.conf` file:

```
iim_ldap.usergroupbynamesearchfilter=(|(&(|(objectclass=groupofuniquenames)
(objectclass=groupofurls)))(cn={0}))(&(objectclass=inetorgperson)(cn={0})))

iim_ldap.groupbrowsefilter=(|(objectclass=groupofuniquenames)(objectclass=g
roupofurls))

iim_ldap.groupclass=groupOfUniqueNames,groupOfURLs
```

The attribute and objectclass names are configurable.By default, the `memberOfUrls` attribute is used as the membership attribute of a dynamic group. If you want to use an attribute name other than `memberOfUrls`, set the `iim_ldap.groupmemberurlattr` option to the attribute name you want to use.

Backing Up Instant Messaging Data

Instant Messaging does not come with any disaster recovery tools. Use your site's backup system to backup the configuration and database directories periodically.

Backup Information

The Instant Messaging information that needs to be backed up are of the following types:

- Configuration Information
- Instant Messaging end user data
- Instant Messenger resources

The configuration information is stored in the configuration directory (*im_cfg_base*). Default paths are as follows:

- Solaris: `/etc/opt/SUNWiim/default/config`
- Linux: `/etc/opt/sun/im/default/config`
- (Optional) If you customized any of the files mentioned in Customizing Instant Messenger, back them up from the resource directory.

The Instant Messaging data is stored in the database directory (*im_db_base*). Defaults for *im_db_base* are as follows:

- Solaris: `/var/opt/SUNWiim/default/db`
- Linux: `/var/opt/sun/im/db`

The Instant Messenger resources must be backed up if they have been customized. The location of the Instant Messenger resources are provided during installation.

Performing a Backup

While the configuration information does not change frequently, the Instant Messaging end-user data changes rapidly and to prevent any loss of end-user data you should back up the Instant Messaging end-user data on a periodic basis. You need to perform the backup before running the installation program and the uninstallation program.

To backup the end user data and the configuration information you do not have to stop the Instant Messaging server as all the disk commits by the server are automatically performed.

Restoring Backup Information

The back up of the end-user data and the configuration information needs to be restored when there is a disk failure and all the end-user data and the configuration information is lost.

➤ To Restore End-user Data from Backup

1. Change to the runtime directory.

   For example:

   `cd` *im_runtime_base*

   The default values for *im_runtime_base* are as follows:

   Solaris: `/var/opt/SUNWiim/default`

   Linux: `/var/opt/sun/im/`

2. Stop the Instant Messaging server, type:

   ```
   imadmin stop
   ```

3. Copy the backed up data to the *im_db_base* directory. Be sure to maintain the directory structure of the backed up data.

4. Verify the permissions and owner of the newly restored data.

   The files should be owned by the Instant Messaging system user. See "Creating a UNIX System User and Group" on page 24 for information on this user. Permissions should be set as follows:

   - Files: `600` (indicating read and write permissions for owner only)
   - Directories: `700` (indicating read, write, and execute permissions for owner only)

   Refer to your operating system documentation for information on changing permissions and owners.

5. Start the Instant Messaging server.

   ```
   imadmin start
   ```

Chapter 5

Managing Instant Messenger

This chapter describes how to customize and administer Instant Messenger in the following sections:

- Configuring Instant Messenger
- Invoking Instant Messenger
- Changing the Codebase
- Changing the Web Container Port
- Customizing Instant Messenger
- Instant Messenger Resource Files
- Modifying How Client Users Search for Contacts
- Administering Conference Rooms and News Channels
- Modifying Instant Messenger Proxy Settings
- Controlling the Exposed Messenger Feature Set
- Instant Messenger Data Stored in the End User’s System
- Using Calendar Pop-up Reminders
- Redeploying Resource Files

Configuring Instant Messenger

There are two ways to invoke and run Instant Messenger:

Using Java Web Start In this configuration, Instant Messenger is launched as an application from the Java Web Start. The browser is no longer necessary once Instant Messenger is launched.

Using the Java Plug-in In this configuration, Instant Messenger is run as a Java applet. To keep the Instant Messenger session active, the browser window from which the applet was launched must remain open and cannot be used to locate any other URL.

For more information on how to configure the Java software that enables Instant Messenger, see Chapter 2, "Setting up and Launching Instant Messenger".

Invoking Instant Messenger

You can invoke Instant Messenger using:

- The `index.html` file that provides you the options to launch both the Java Web Start and Java Plug-in versions of Instant Messenger. This file also contains links to Instant Messenger documentation.
- The web page that you have designed with a link to Instant Messenger.
- A direct URL for either the `im.html` or `im.jnlp` files.

To Invoke Instant Messenger

Use the following URL to invoke the Instant Messenger.

`http://`*webserver***`:`***webserverport***`/`***path***`/`***filename*

In this URL,

webserver	Specifies the name of the web container where you have installed the Instant Messenger resources.
webserverport	(Optional) Specifies the web container port. The default value is `80`.
path	(Optional) Specifies the directory where the client files are installed. If the default is selected during the installation, then no subdirectory is required to store the client files.
filename	Specifies the Instant Messenger file to use: `index.html` - This file is provided with the product. The file contains links to `im.jnlp` and `im.html` which launch both the Java Web Start and Java Plug-in versions of Instant Messenger. `im.jnlp` - The jnlp file to launch only the Java Web Start version of Instant Messenger. `im.html` - The page to launch only the Java Plug-in version of Instant Messenger.

You can also do the following:

- Add the URL to your favorites.
- Launch the application using the Java Web Start icon on your desktop.
- Create and use a desktop shortcut.

  You can do this one of two ways:

  - Create a shortcut using Java Web Start.
  - Create a shortcut manually and set the target value as follows:

    `javaws_cmd` *jnlp_URL*

    Where *jnlp_URL* is the URL to the `im.jnlp` file.
- On Solaris, to invoke Instant Messenger from the command-line, type:

  `javaws_cmd` *URL*

Changing the Codebase

The codebase is the URL from which Instant Messenger accesses resources, including the start page for initial downloads of the Instant Messaging client. This URL is defined during postinstallation configuration when the resource files are deployed by the `configure` utility. If you change any portion of the URL used to access Instant Messenger resources including the web container port number you need to update the codebase.

If you want to change the codebase after you have deployed the resource files you will need to:

- Modify the template files to point to the new URL. See "To Change the Codebase in the Resource Templates" in this section for instructions.
- Re-run the configure utility, selecting the "Messenger Resources" component only when prompted for which components you want to configure. See "Configuring Instant Messaging After Installing or Upgrading" on page 25 for instructions.
- Redeploy the resource files. See "Redeploying Resource Files" on page 101 for instructions.

➤ To Change the Codebase in the Resource Templates

- Edit each of the template files in the *im_svr_base*/`html` directory with the new URL.

  Template files are named `*.template`. See Table 5-1 on page 77 for a complete list of template files.

Changing the Web Container Port

If you change any portion of the URL used to access Instant Messenger resources including the web container port number you need to update the codebase. See "Changing the Codebase" on page 76 for instructions.

Customizing Instant Messenger

Instant Messenger is customizable. HTML and JNLP files can be customized to suit an organization's specific needs. If you want to customize the resource files for your deployment, you should run `configure` utility (if you haven't already done so after installing), customize the files, then redeploy the resource files. You need to run configure first because the configure utility creates some of the index and `.jnlp` files that you can customize. (See "Redeploying Resource Files" on page 101 for redeployment instructions.)

You can customize the Instant Messenger to meet your requirements in the following ways:

- Customizing the index.html and im.html Files (Only LDAP Deployments)
- Customizing the Application (Java Web Start)
- Customizing User Name Display

This section describes the Instant Messaging server files you can modify to customize Instant Messenger. The files that you can customize are all located in the `html` directory. For example, on Solaris the HTML files are located in the resource directory.

Instant Messenger Resource Files

The Instant Messenger resource files are located within the directory referred to as the resource directory.

Table 5-1 contains the list of Instant Messenger files in the resource directory (*im_svr_base*/`html`). It also contains the description and the customization information of these files. Within the resource directory, the locale subdirectory is represented generically in a directory path as *lang*, but specifically as abbreviations of languages, such as, `en_US`, `jp`, and `fr_FR`.

Table 5-1 Instant Messenger Files

File	Description	Customizable?
lang/`im.html`	The initial page that launches the Java Plug-in version of Instant Messenger.	Yes.

Table 5-1 Instant Messenger Files *(Continued)*

File	Description	Customizable?
im.html.template	The template version of im.html.	No. This file is used by the installation program to generate the im.html file.
imdesktop.jar	A client jar file, downloaded by im.html or im.jnlp files.	No.
lang/im.jnlp	The jnlp file to launch Java Web Start version of Instant Messenger.	Yes.
im.jnlp.template	The template version of im.jnlp.	No.
imjni.jar	A client jar file, downloaded by im.html or im.jnlp.	No.
messenger.jar	The main client jar file, downloaded by im.html or im.jnlp.	No
icalendar.jar	The icalendar parser used to process calendar reminders.	No
imnet.jar	A client jar file, downloaded by im.html or im.jnlp.	No.
lang/imbrand.jar	This file contains customizable properties, stylesheets, images, backgrounds, and audio files.	Yes
lang/imssl.html	The Initial page that launches Java Plug-in version of Instant Messenger. It is used for running SSL between the client and the multiplexor.	Yes.
imssl.html.template	The template version of imssl.html	No.
lang/imssl.jnlp	This file launches Java Web Start version of Instant Messenger. This file is used for running SSL between the client and the multiplexor.	Yes.
imssl.jnlp.template	The Template version of imssl.jnlp file.	No.
jnlpLaunch.jsp	If an end user is already logged onto Sun Java System Access Manager, then this file can be used to allow single sign-on and to launch Instant Messenger using Java Web Start.	Yes

Table 5-1 Instant Messenger Files *(Continued)*

File	Description	Customizable?
pluginLaunch.jsp	If an end user is already logged onto Sun Java System Access Manager, then this file can be used to allow single sign-on and to launch Instant Messenger using Java Plug-in.	Yes
index.html	The splash page for an LDAP deployment. It contains links to im.html and im.jnlp, as well as documentation links to windows.htm, solaris.htm, and quickref.htm. You can customize this page for your site's requirement.	Yes.
index.html.template	The template version of index.html.	No.
lang/imhelp/SunONE.jpg	The image used by quickref.htm, solaris.htm, and windows.htm.	Can be replaced.
quickref.html solaris.html windows.html	Located in lang/imhelp/, they provide documentation on getting started with Instant Messenger.	Yes.
lang/imhelp	Instant Messenger Online Help directory.	No.
imwebex.jar		
msgrinstall.jar		

Customizing the index.html and im.html Files (Only LDAP Deployments)

The Instant Messenger allows you to modify the "static" portion of the `index.html` and `im.html` files to produce a fully customized user interface. These HTML files contain both text and markups describing how the text is formatted and handled. Markup is implemented through a set of tags, which specify formats for headers, indents, font size, and font style.

Some of the page elements that can be modified are:

- Images and Banner
- Text on screen including title and field labels
- Background schemes

The `index.html` file launches both the Instant Messenger applet and the Java Web Start application. If you are running the Instant Messenger applet, modify the `im.html` file. The `im.html` file is called by `index.html`, and invokes the Instant Messenger applet. The `im.html` file is generated during the installation and contains an applet argument that points to the multiplexor.

NOTE The argument "`<PARAM NAME="server" VALUE="servername">`" represents the Instant Messaging multiplexor and its port in the `im.html` file. If you change the `iim_mux.listenport` parameter's default value, you need to change the *servername* value to *host.domain*`:`*port*.

Launching Instant Messenger Using Sun Java System Access Manager SSO

To launch the Instant Messenger client using single sign-on (SSO) with Sun Java System Access Manager use `IMLaunch.jsp`. This file is in the resource directory.

Sun Java System Access Manager and the Instant Messenger client must be configured to use the same web container to enable SSO.

To launch the Instant Messaging server enter the following in the browser:

codebase`/IMLaunch.jsp?server=`*multiplexor-hostname*`:`*muliplexor-port*

or

codebase`/IMLaunch.jsp?server=www.example.com:5222`

where,

codebase is the codebase from which the Instant Messenger resources are downloaded. For example, `http://www.example.com`.

multiplexor-hostname is the name of the mulitplexor. For example, `http://www.company22.com`.

muliplexor-port is the port number on which the multiplexor listens for incoming client requests. For example, 5222.

`IMLaunch.jsp` is used for launching Instant Messenger through either Java Web Start or Java Plug-in.

Customizing the Application (Java Web Start)

If you are running Instant Messenger using Java Web Start, you can modify the `im.jnlp`, `imres.jnlp`, and `imres.jar` files to customize the user interface. The following are modifications that can be made to these HTML files:

- `im.jnlp` - this file invokes the Java Web Start version of the Instant Messenger application. You can modify the codebase, title, vendor, and descriptions in the file.

  Table 5-2 shows a sample `im.jnlp` file with the HTML code that can be customized in bold typeface.

Table 5-2 Sample `im.jnlp` file.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Instant Messenger -->
<jnlp
  spec="1.0+"
  codebase="http://im.i-zed.com:80/im"
  href="en/im.jnlp">
  <information>
    <title>Instant Messaging</title>
    <vendor>I-Zed.com</vendor>
    <homepage href="http://www.I-zed.com/"/>
    <description>I-Zed’s Sun Java System Instant Messenger/description>
    <description kind="short">Instant Messenger/description>
    <icon href="CompanyLogo.gif"/>
    <offline-allowed/>
  </information>
  <security>
    <all-permissions/>
  </security>
  <resources>
    <j2se version="1.3+">
      <resources>
        <jar href="en/imres.jar"/>
        <jar href="en/imbrand.jar"/>
      </resources>
    </j2se>
    <jar href="messenger.jar"/>
    <jar href="imdesktop.jar"/>
    <jar href="imnet.jar"/>
    <jar href="icalendar.jar"/>
    <nativelib href="imjni.jar"/>
  </resources>
  <application-desc main-class="com.iplanet.im.client.iIM">
    <argument>server=im.i-zed.com:45222</argument>
    <argument>help_codebase=http://im.i-zed.com:80/im/en</argument>
  </application-desc>
</jnlp>
```

NOTE In the `im.jnlp file`, the argument `<argument>`*servername*`</argument>` represents the Instant Messaging multiplexor host and port. If you change the default value of `iim_mux.listenport` parameter, you need to change the *servername* value to *host.domain*:*port*.

- `imbrand.jar` - This file contains the image and audio files, and the properties that can be customized. You need Java Developers Kit 1.3(JDK) to extract the contents from the imres.jar file using the `jar` command. For more information on the `imbrand.jar` file contents, see Table 5-3 on page 83.

  The following is the syntax for the `jar` command:

  ```
  jar xvf imbrand.jar
  ```

  This command creates a directory tree where the resource files are copied. This directory structure has to be maintained when you modify the individual files in the `jar` file.

  You can substitute your version of `.gif` files or `.wav` files, without changing the file names and then place the changed files back to the directory using the following `jar` command:

  ```
  jar -uf imbrand.jar com/Sun/im/client/images/*.gif
  ```

  This command updates the `imbrand.jar` file with the modified `.gif` files. The same is possible with the audio files (`.wav` files).

Contents Listing of imbrand.jar

Table 5-3 lists the files in the `imbrand.jar` file and their description. The `imbrand.jar` file contains the image and audio files that can be used to re-brand Instant Messenger.

Table 5-3 imbrand.jar Contents

File Name	Description
Configuration Files	
`brand.properties`	
`chat-styles.css`	
Emoticons	
`emo_alarm.png`	Shows alarm emotion graphically

Table 5-3 imbrand.jar Contents *(Continued)*

File Name	Description
emo_angel.png	Shows angelic emotion graphically
emo_angry.png	Shows angry emotion graphically.
emo_balloons.png	Graphic depiction of a bunch of balloons.
emo_beermug.png	Graphic depiction of a mug of beer.
emo_cake.png	Graphic depiction of a birthday cake.
emo_calendar.png	Graphic depiction of a calendar.
emo_canworms.png	Graphic depiction of a can of worms.
emo_clown.png	Graphic depiction of a clown’s head.
emo_cool.png	Shows cool emotion graphically.
emo_dead.png	Indicates dead graphically.
emo_devil.png	Shows devilish emotion graphically.
emo_dont-tell.png	Indicates a request for secrecy graphically.
emo_embarrassed.png	Shows embarrassed emotion graphically.
emo_exclamation.png	Graphic depiction of an exclamation point.
emo_flower.png	Graphic depiction of a flower.
emo_ghost.png	Graphic depiction of a ghost.
emo_goldstar.png	Graphic depiction of a gold star.
emo_grin.png	Shows a grin graphically.
emo_kiss.png	Shows a kiss graphically.
emo_laughing.png	Show laugh emotion graphically.
emo_lifepreserver.png	Graphic depiction of a life preserver.
emo_lightning.png	Graphic depiction of a thunder cloud and lightning bolt.
emo_lovestruck.png	An emoticon used to show love emotion graphically
emo_martini.png	Graphic depiction of a martini glass.
emo_money.png	Graphic depiction of stacks of coins.
emo_musicnote.png	Graphic depiction of a musical note.
emo_nerd.png	Graphic depiction of a nerd.
emo_nottalking.png	Shows a turned-away countenance graphically.
emo_phone.png	Graphic depiction of a phone receiver.
emo_present.png	Graphic depiction of a wrapped gift.

Table 5-3 imbrand.jar Contents *(Continued)*

File Name	Description
emo_psychoknife.png	Graphic depiction of a knife.
emo_rathole.png	Graphic depiction of a rat hole.
emo_sad.png	Shows sad emotion graphically.
emo_sick.png	Shows illness graphically.
emo_sleep.png	Shows sleepiness graphically.
emo_smiley.png	Shows a smile graphically.
emo_straightfaced.png	Graphic depiction of a straight-faced person.
emo_sunshining.png	Graphic depiction of a sun.
emo_surprised.png	Shows suprise graphically.
emo_tongue-out.png	Graphic depiction of a person sticking out his tongue.
emo_violin.png	Graphic depiction of a violin.
emo_whatever.png	Shows indifference or disdain graphically.
Application Icons - Windows	
im_app_icon_16.png	Title bar icon for Windows.
im_app_icon_24.png	Title bar icon for Windows.
tray_icon.ico	System tray icon for Windows.
Application Icons - All Platforms	
logo_login_footer.png	Logo displayed at the bottom of the Login dialog box.
logo_register.png	Logo displayed on the Register dialog box.
logo_sun.png	Sun logo displayed on the Login dialog box.
Toolbar Icons	
tb_addcontacts.png	Graphic for the Add Contacts button.
tb_alert.png	Graphic for the Send Alert button.
tb_chat.png	Graphic for the Chat With Users button.
tb_conf.png	Graphic for the Add Conferences button.
Contact List Icons	
cl_folder_closed.png	Shows a closed folder graphically.
cl_folder_open.png	Shows an open folder graphically.
Presence Icons - Contact List	

Table 5-3 imbrand.jar Contents *(Continued)*

File Name	Description
`cl_activeconf.png`	Icon displayed to indicate an active conference that appears in the Contact List.
`cl_away.png`	Icon for away status that appears in the Contact List.
`cl_dnd.png`	
`cl_idle.png`	Icon displayed to show idle status that appears in the Contact List.
`cl_inactiveconf.png`	Icon displayed to indicate an inactive conference that appears in the Contact List.
`cl_offline.png`	Icon for offline status that appears in the Contact List.
`cl_online.png`	Icon for online status that appears in the Contact List.
`cl_pending.png`	Icon that indicates pending status that appears in the Contact List.
Presence Icons - Status Bar	
`sb_away.png`	Icon for away status that appears in the Status Bar.
`sb_dnd.png`	
`sb_idle.png`	Icon for idle status that appears in the Status Bar.
`sb_offline.png`	Icon for offline status that appears in the Status Bar.
`sb_online.png`	Icon for online status that appears in the Status Bar.
Background Configuration Files	
`bgstyles.properties`	Configuration file used to extend the background set.

Table 5-3 imbrand.jar Contents *(Continued)*

File Name	Description
Backgrounds and Background Swatches for the Palette	
bgplt_tex_blue.gif	bgplt_tex_weave_purple.gif
bgplt_tex_brown.gif	bgplt_tex_weave_ruby.gif
bgplt_tex_bubble_blue.gif	bgplt_tex_white.gif
bgplt_tex_bubble_brown.gif	bg_tex_bubble_blue.gif
bgplt_tex_bubble_green.gif	bg_tex_bubble_brown.gif
bgplt_tex_bubble_grey.gif	bg_tex_bubble_green.gif
bgplt_tex_bubble_orange.gif	bg_tex_bubble_grey.gif
bgplt_tex_bubble_purple.gif	bg_tex_bubble_orange.gif
bgplt_tex_bubble_ruby.gif	bg_tex_bubble_purple.gif
bgplt_tex_crackle_blue.gif	bg_tex_bubble_ruby.gif
bgplt_tex_crackle_green1.gif	bg_tex_crackle_blue.gif
bgplt_tex_crackle_grey.gif	bg_tex_crackle_green1.gif
bgplt_tex_crackle_olive.gif	bg_tex_crackle_grey.gif
bgplt_tex_crackle_orange.gif	bg_tex_crackle_olive.gif
bgplt_tex_crackle_purple.gif	bg_tex_crackle_orange.gif
bgplt_tex_crackle_ruby.gif	bg_tex_crackle_purple.gif
bgplt_tex_gradation_blue.gif	bg_tex_crackle_ruby.gif
bgplt_tex_gradation_brown.gif	bg_tex_gradation_blue.gif
bgplt_tex_gradation_green.gif	bg_tex_gradation_brown.gif
bgplt_tex_gradation_grey.gif	bg_tex_gradation_green.gif
bgplt_tex_gradation_orange.gif	bg_tex_gradation_grey.gif
bgplt_tex_gradation_purple.gif	bg_tex_gradation_orange.gif
bgplt_tex_gradation_ruby.gif	bg_tex_gradation_purple.gif
bgplt_tex_green.gif	bg_tex_gradation_ruby.gif
bgplt_tex_orange.gif	bg_tex_weave_blue.gif
bgplt_tex_pink.gif	bg_tex_weave_brown.gif
bgplt_tex_purple.gif	bg_tex_weave_green.gif
bgplt_tex_weave_blue.gif	bg_tex_weave_grey.gif
bgplt_tex_weave_brown.gif	bg_tex_weave_orange.gif
bgplt_tex_weave_green.gif	bg_tex_weave_purple.gif
bgplt_tex_weave_grey.gif	bg_tex_weave_ruby.gif
bgplt_tex_weave_orange.gif	

Table 5-3 imbrand.jar Contents *(Continued)*

File Name	Description
Sounds	
`alert.wav`	Alert sound.
`alerttpc.wav`	Alert sound.
`away.wav`	Sound used when you change your status to away.
`receive.wav`	Sound used when you receive a message.
`send.wav`	Sound used when you send a message.
`soundoff.wav`	Sound used when you turn the sound off.
`soundon.wav`	Sound used when you turn the sound on.

Rebranding Instant Messenger

The `imbrand.jar` file contains all images and the properties that control the look and feel of the Instant Messenger. You can customize the appearance of the Instant Messenger by modifying the images and the properties in the `imbrand.jar` file.

➤ To Rebrand Instant Messenger

1. Copy `imbrand.jar` file to a working directory. For example:

   `cp` *im_svr_base*`/html/`*lang*`/imbrand.jar` *working_directory*

2. Change to the working directory.

   `cd` *working_directory*

3. Extract the `imbrand.jar` file.

   ```
   jar xf imbrand.jar
   ```

   This command creates a directory tree where the resource files are copied. This directory structure has to be maintained when you modify the individual files in the `jar` file.

   Alternatively, you can extract a single file included in `imbrand.jar` and put it under the directory structure you specify. For example, to extract only `brand.properties`, use the following command:

   ```
   jar xf imbrand.jar com/sun/im/desktop/brand/brand.properties
   ```

4. Update the `imbrand.jar` file with the modified `.gif`, `.wav`, and `.properties` files.

   You can update all the files in imbrand.jar as follows:

   ```
   jar cf imbrand.jar .
   ```

   To update the `imbrand.jar` file with a single modified file, use the following command:

   ```
   jar uf imbrand.jar com/sun/im/desktop/brand/<filename>
   ```

   Where <*filename*> is the name of the file included in `imbrand.jar`, for example, `brand.properties`.

5. Copy the `imbrand.jar` file to the resource directory. For example:

   ```
   cp imbrand.jar im_svr_base/html/lang/ .
   ```

NOTE If multiple locales are supported, the procedure for re-branding Instant Messenger should be followed for every supported locale.

Customizing User Name Display

The User Name display can be customized in the tooltip and the search results.

Customizing User Name Display in Search Results

When two end users have the same first name and last name, it is impossible to know which end user has to be added to the contact list. You can customize the Instant Messenger to display more information in the search results for the user search. For displaying more information in the user search results, in the `imbrand.jar` file you need to add `dialogs.searchresults.format` attribute to the `brand.properties` file at:

```
com/sun/im/desktop/brand/
```

For more information on how to modify `imbrand.jar`, see Customizing the Application (Java Web Start).

More information can be displayed in the user search results by including additional LDAP attribute values in the `dialogs.searchresults.format` attribute.

The LDAP attributes are specified in the following format:

```
${attr:attribute-name}
```

The following example shows the LDAP attribute in `dialogs.searchresults.format` attribute:

```
dialogs.searchresults.format=(${attr:title})
```

To use arbitrary attributes from the LDAP user entry, the list of these custom attributes needs to be specified in the server configuration file `iim.conf`. These custom attributes need to be specified as values for the attribute `iim_ldap.userattributes`.

The following example shows the `iim_ldap.userattributes` with the list custom attributes as value:

```
iim_ldap.userattributes=title,department,telephonenumber
```

Customizing User Name Display in Tooltip

You can customize the Instant Messenger to display additional information in the Contact tooltip.

For example, to display the phone number of the Contact when the mouse is placed over the Contact:

1. Change to the following directory:

   ```
   com/sun/im/client/
   ```

2. Open the `brand.properties` file.
3. Add the `contact.tooltip.format.html` attribute to the file.
4. Save the changes to the file
5. Change to the following directory.

   `cd` *im_svr_base*`/html`

6. Add the `contact.tooltip.format.html` attribute and the `telephonenumber` attribute as its value in the HTML code of the `imbrand.jar` file:

   ```
   contact.tooltip.format.html=mailto: ${attr:mail} tel:
   ${attr:telephonenumber}
   ```

For information on customizing the `imbrand.jar` file, see Customizing the Application (Java Web Start).

Modifying How Client Users Search for Contacts

By default the “commonname” or “cn” LDAP attribute is used to represent a user’s display name and as a search attribute for users. If you would like to use the “nickname” LDAP attribute instead, you can do so by setting the following parameter in `iim.conf`:

```
iim_ldap.userdisplay=nickname
```

To allow the user to search by either commonname or nickname, modify the `iim_ldap.usergroupbynamesearchfilter` attribute. This specifies the LDAP search used when searching for user or groups. It uses the standard LDAP filter syntax. You can modify it to allow more complex searches. See your Directory Server documentation for more information on modifying search strings.

Administering Conference Rooms and News Channels

Listed below are tasks that you can perform in Instant Messenger to administer the conference rooms and the news channels. For more information on performing these tasks, see the Online Help.

- Administering conference rooms
- Administering and managing news channels
- Assigning conference room access levels to end users
- Assigning news channel access levels to end users
- Assigning end users to conference rooms
- Assigning end users to news channels (subscribing)
- Creating new conference rooms
- Creating new news channels
- Configuring end user settings
- Deleting conference rooms
- Deleting messages from news channels
- Deleting news channels
- Posting messages in news channels

- Removing end users from conference rooms
- Removing end users from news channels

Granting End Users the Privilege to Create Conference Rooms and News Channels

The administrator can create conference rooms and news channels for end users. However, with the proper privileges, end users can do this also. For more information about adding policies to give end users access to create conference rooms and news channels, see Chapter 6, "Managing Instant Messaging and Presence Policies" on page 103. End users who create a conference room or a news channel by default have Manage access, enabling them to administer the conference room or the news channel.

Modifying Instant Messenger Proxy Settings

Instant Messaging messages can contain embedded URLs. For example, `http://stocks.yahoo.com?id=sunw`. If you are using proxy servers, you need to resolve such embedded URLs by modifying the Instant Messenger proxy settings in the Java Web Start configuration.

This is likely to happen if your organization has a firewall, and you need to go through the proxy server before connecting your client hosts to internet, and if Java Web Start has not been configured with the right proxy settings.

To Modify Instant Messenger Proxy Settings

Java Web Start can automatically configure the proxy settings by querying the system or the default browser. However, it is not possible for the Java Web Start to automatically configure these settings if the proxy settings are configured using a JavaScript file.

➤ To Set Proxy Settings Manually

1. Invoke Java Web Start.
2. From the File menu, choose Preferences.
3. Select Manual option in the Preferences dialog.

4. Enter the following details:

   HTTP Proxy. Enter the Name or the IP address of the proxy server.

   HTTP Port. Enter the port number of the proxy server.

   No Proxy_Hosts. Enter the name of any domain that you can connect directly, bypassing the proxy server. Use commas to separate multiple host names.

5. Click OK to save the proxy settings.

Controlling the Exposed Messenger Feature Set

The exposed feature set of Instant Messenger can be controlled by the administrator by configuring the Instant Messaging applet parameters in the applet descriptor files.

Table 5-4 shows the Instant Messenger applet parameters in the applet descriptor files. It also contains the description and the default values of these parameters.

Table 5-4 Instant Messenger Applet Parameters

Parameter	Default Value	Description
`server`	`127.0.0.1`	The Instant Messaging server host and port.
`debug`	`FALSE`	If this parameter is set to true, the applet records all the task performed on java console.
`uid`		This parameter is used for SSO.
`token`		This parameter contains the SSO token and is used for auto-logon.
`secure`	`FALSE`	Indicates to the Instant Messenger that it is run in SRA mode. It displays a security indicator.
`usessl`	`FALSE`	Tells Instant Messenger to use SSL when connecting to server.
`allow_alert_only`	`FALSE`	Tells Instant Messenger to let end user display neither the contact list nor the news channel. This parameter is used in `CHAT` and `POPUP` flavors.
`allow_file_transfer`	`TRUE`	Allows file attachment and transfer.

Table 5-4 Instant Messenger Applet Parameters *(Continued)*

Parameter	Default Value	Description
`enable_moderator`	`TRUE`	If set to true, enables the moderated conference feature.
`messenger_bean`		This parameter contains a list of messenger beans to be used. You can enter multiple factory class names with each separated by a comma.
domain	null	This parameter is used in multidomain Sun Java System Access Manager deployments. The value of this parameter should be the logical domain name of the organization in which this end user is present.
`gateway_url`	null	This parameter contains the URL of the gateway component of portal SRA.

Instant Messenger Data Stored in the End User's System

Instant Messenger caches a limited amount of information on the end user's system for auto-login. This information is located at:

home_directory`/.sunmsgr`

home_directory is the end user's home directory. The home directory of the end user can be obtained from the `user.home` parameter in the Java system property.

Table 5-5 shows the directories and files containing the cached data. It also contains the description of the files and the directories.

Table 5-5 Cached Data Directory and Files

File/Directory Name	Type	Description
`.sunmsgr/messenger.properties`	file	The file containing the auto-logon properties
`.sunmsgr/`*<user_domain>*	directory	Directory containing data specific to a particular {log-in name, domain name} combination.

Table 5-5 Cached Data Directory and Files *(Continued)*

File/Directory Name	Type	Description
home_directory/sunmsgr/<*user_domain*>/messenger.properties	file	This file contains auto-logon options specific to particular <*user_domain*>. This file is not used.
home_directory/sunmsgr/<*user_domain*>/messages/	directory	This directory contains cached messages. This directory is not used.

Table 5-6 shows the auto-logon properties for Instant Messaging. It also contains the description and the default values of these properties.

Table 5-6 Auto-logon Properties

Parameter	Default Value	Description
client.password.encoded	false	Determines whether or not the user password is encoded (for use with SSO). If the value for this parameter is true, the encoded password is stored as the value for the net.password parameter.
net.server	127.0.0.1	Instant Messaging server host name and port.
net.server.*n* (Where *n* is a digit used to distinguish one entry from another)		The secondary servers' host names and port numbers.
net.user		The default user id
net.password		The encoded user password that enables auto-logon.

Using Calendar Pop-up Reminders

Instant Messaging is integrated with Sun Java System Calendar Server to provide automatic pop-up reminders to Instant Messenger users for both calendar events and tasks.

This section contains the following topics:

- Pop-up Reminders Overview
- Configuring Instant Messaging Pop-ups
- Administering the Calendar Agent

Pop-up Reminders Overview

This section contains the following topics:

- “Pop-up Reminders Operation” on page 96
- “Pop-up Reminders Architectural Flow” on page 97

Pop-up Reminders Operation

Users can receive Instant Messenger pop-up reminders for upcoming events and tasks on their calendars. To enable these pop-up reminders, the following must occur:

- The administrator must configure the Calendar server and the Instant Messaging server to allow pop-up notifications.
- The end user must specify email reminders in the Options tab of either Calendar Express or Communications Express, which sets an alarm in the Event Notification System.
- The end user must enable calendar reminders in Instant Messenger.

With pop-ups enabled, when an impending event or task nears, the alarm set in the Event Notification System causes Calendar Server to send an email notification and Instant Messaging to display a pop-up reminder.

Pop-up Reminders Architectural Flow

If configured, Instant Messaging pop-up reminders follow this architectural flow:

1. The Instant Messaging JMS subscriber subscribes to Calendar server events and notifications in the Event Notification Service (ENS).
2. Calendar server publishes an event or task notification in `text/xml` or `text/calendar` format to ENS.
3. The Instant Messaging JMS subscriber receives the calendar event or task notification and then generates a message in `text/calendar` format.
4. The Instant Messaging server sends the message to the calendar owner, if the end user is online.
5. If the recipient is available, Instant Messenger generates an HTML pop-up reminder on the end user's desktop based on the message.

   If the recipient is not available, the Instant Messaging server discards the message.

Configuring Instant Messaging Pop-ups

This section includes the following configuration instructions:

- To Configure Instant Messaging Server
- To Configure Calendar Server
- To Configure Instant Messenger

➤ To Configure Instant Messaging Server

1. Install the new package `SUNWiimag`.

2. Edit one or more of the parameters in the `iim.conf` file as shown in Table 5-7.

   The parameter values shown assume you want pop-up reminders for both events and tasks. If these parameters do not already exist in your `iim.conf` file, add them.

Table 5-7 iim.conf Parameters for Configuring Calendar Pop-ups

Parameter	Description and Appropriate Value to Use
JMS Consumers Section	
`jms.consumers`	Name of alarm. Set the value to: `cal_reminder`
`jms.consumer.cal_reminder.destination`	Destination of the alarm. This must be the same as the value of the `caldb.serveralarms.url` configuration parameter in the `ics.conf` file. For example, `enp:///ics/customalarm`
`jms.consumer.cal_reminder.provider`	The name of the provider. Set to `ens`. This must be the same as the name in `jms.providers` in the JMS Providers section.
`jms.consumer.cal_reminder.type`	The type of alarm to set. Set the value to: `topic`
`jms.consumer.cal_reminder.param`	The alarm parameter. Set the value as follows including the quotes: `"eventtype=calendar.alarm"`
`jms.consumer.cal_reminder.factory`	A listener that registers itself for the new calendar reminder messages. Set the value to: `com.iplanet.im.server.JMSCalendarMessageListener`
JMS Providers Section	
`jms.providers`	The name of the provider. Set value to `ens`. This must be the same as the value listed in the JMS Consumers Section for jms.consumer.cal_reminder.provider.
`jms.provider.ens.broker`	Hostname of the ENS and the port number on which the ENS listens for incoming requests. Set to the port specified in the `ics.conf` file parameter `service.ens.port`. The default is 57997. For example: `jms.provider.ens.broker=cal.example.com:57997`

Table 5-7 iim.conf Parameters for Configuring Calendar Pop-ups *(Continued)*

Parameter	Description and Appropriate Value to Use
`jms.provider.ens.factory`	Factory class used for creating the topic connection objects. Set the value to: `com.iplanet.ens.jms.EnsTopicConnFactory`
Instant Messaging General Parameters	
`iim_agent.enable`	Enables the Calendar agent. Set the value as follows including the quotes: `iim_agent.enable="true"`
`iim_agent.agent-calendar.enable`	Loads a component that enables the Calendar agent. Set the value as follows including the quotes: `iim_agent.agent-calendar.enable="true"`
`agent-calendar.jid`	The JID of the Calendar agent. Set this value as follows: `agent-calendar.jid=calimbot.`*server*`.`*domain*
`agent-calendar.password`	The Calendar agent password. Set this value as follows: `agent-calendar.password=`*password*
`iim_server.components`	Set this value as follows: `iim_server.components=agent-calendar`

3. Start the Calendar agent using `imadmin`.

   ```
   imadmin start agent-calendar
   ```

   The `imadmin` command-line utility is located in the following directory:

   im_svr_base`/sbin`

   Where *im_svr_base* is the directory in which you installed Instant Messaging.

➤ To Configure Calendar Server

1. Confirm that the `ics.conf` parameters shown in Table 5-8 have the values shown. If they do not, perform the following steps to make them conform.
2. Log in to the Calendar server host as an administrator with permission to change the configuration.
3. Change to the `/etc/opt/SUNWics5/cal/config` directory.
4. Save your old `ics.conf` file by copying and renaming it.
5. Edit the parameters in Table 5-8 to the values shown.

Table 5-8 ics.conf Parameters for Configuring Calendar Pop-ups

Parameter	Description and Default Value
`caldb.serveralarms`	Enables calendar alarms to be queued. The default is `"yes"` (enabled).
`caldb.serveralarms.contenttype`	Output format for alarm content. The default is `"text/xml"`.
`caldb.serveralarms.dispatch`	Enables calendar alarms to be dispatched. The default is `"yes"`.
`caldb.serveralarms.dispatchtype`	The type of server alarm to dispatch. The default is `"ens"`.
`caldb.serveralarms.url`	This is the URL for alarm retrieving alarm contents. The default is `"enp:///ics/customalarm"`.

6. Save the file as `ics.conf`.

7. Restart Calendar server.

   cal_svr_base`/SUNWics5/cal/sbin/start-cal`

   Where *cal_svr_base* is the directory in which you installed Sun Java System Calendar Server.

➤ To Configure Instant Messenger

1. On the Instant Messenger Main window, from the Tools menu, select Settings.

2. On the Settings window, click the Alerts tab.

3. Check the Show Calendar Reminders option.

4. Click OK.

   Users can now receive Calendar pop-ups through Instant Messenger while they are online.

Administering the Calendar Agent

The Calendar agent is an Instant Messaging component that provides pop-up functionality to Calendar and Instant Messaging users. Using tools provided with Instant Messaging, you can start, stop, restart, or check the status of the Calendar agent as well as monitor its activity through log files. See "Stopping, Starting, and Refreshing Instant Messaging Components" on page 45 for information on administering the Calendar agent component. Also see "Managing Logging" on page 50 for information about Calendar agent logs.

Redeploying Resource Files

If you are using Sun Java™ System Application Server or Sun Java™ System Web Server, and you make changes to the resource files after you run the `configure` utility as a result of site changes or customization, you need to redeploy the files to the web container.

➤ To Redeploy Resource Files to Sun Java System Access Manager or Sun Java System Web Server

- Run the `redeployApp` program from the command line.

  im_svr_base`/html/redeployApp`

  Where *im_svr_base* is the directory in which you installed Instant Messaging.

See the documentation for your web container for additional information.

Chapter 6

Managing Instant Messaging and Presence Policies

Instant Messaging provides various functional features such as chat, conferencing, polls, presence access, etc. A policy describes a set of access control privileges that can be associated with these features. In turn, end users and groups can be assigned to policies according to the needs of an organization.

This chapter describes how to define and use policies to manage the access that end users and administrators have to the Instant Messaging server features and privileges:

- Overview of Privacy, Security, and Site Policies
- Methods for Controlling End User and Administrator Privileges
- Managing Policies Using Access Control Files
- Managing Policies using Sun Java System Access Manager

Overview of Privacy, Security, and Site Policies

Instant Messaging provides the ability to control access to Instant Messaging features and preserve end-user privacy.

Site Policies

Site policies specify end-user access to specific functionality in Instant Messaging. It specifies:

- Ability to access the presence status of other end users
- Ability to send alerts to other end users
- Ability to save properties on the server
- Ability to create and manage conference rooms
- Ability to create and manage news channels

The Instant Messaging administrator has access to all Instant Messaging features. The administrator has `MANAGE` access to all conference rooms and news channels, can view presence information of any end user, and can view and modify properties such as Contact Lists and Instant Messenger Settings of any end user. The site policy settings have no impact on the administrator's privileges.

By default, the end user is provided with the privileges to access the presence status of other end users, send alerts to end users, and save properties to the server. In most of the deployments, the default values are not changed. These default values need to be changed when Instant Messaging is used exclusively for the pop-up functionality.

When Instant Messaging is used exclusively for the pop-up functionality, the end user will not be provided with the access privileges to presence information, chat, and news features.

NOTE Although certain privileges can be set globally, the administrator can also define exceptions for these privileges. For example, the administrator can deny certain default privileges to select end users, roles, or groups.

For more information on configuring site policies, see "Managing Instant Messaging and Presence Policies" on page 103.

Conference Room and News Channel Access Controls

End users can have the following access privileges on Conference rooms and News channels:

- `MANAGE` - full access, which includes the ability to set the conference room or the news channel privilege for other end users.
- `WRITE` - privilege to add contents to the conference room or the news channel.
- `READ` - privilege to read the conference room or the news channel contents.
- `NONE` - no access privileges.

End users with the `MANAGE` privilege can set the default privilege level for all the other end users. These end users can also define the exception rules to grant an access level that is different from the default access level permission given to specific end users or groups.

NOTE Setting the `WRITE` privilege, grants the end users the `READ` privilege.

User Privacy

End users can specify if other end users can see their presence or not. By default, all end users can access the presence information of another end user. End users can also set exceptions for denying this access to certain end user and groups.

If an end user has denied other end users from accessing the end user’s presence status, then that end user’s availability status appears as offline in others contact lists. No alerts or chat invitations can be sent to an end user whose presence status is offline.

User privacy can be configured using the User Settings window in the Instant Messenger. For more information on configuring user privacy, see *Instant Messenger Online Help.*

Methods for Controlling End User and Administrator Privileges

Different sites using Instant Messaging server have different needs in terms of enabling and restricting the type of access end users have to the Instant Messaging service. The process of controlling end user and administrator Instant Messaging server features and privileges is referred to as policy management. There are two methods of policy management available: through access control files or through Sun Java System Access Manager.

Introduction to Managing Policies Using Access Control Files

The access control file method for managing policies allows you to adjust end-user privileges in the following areas: news channel management, conference room management, the ability to change preferences in the User Settings dialog, and ability to send alerts. It also allows specific end users to be assigned as system administrators.

Introduction to Managing Policies Using Sun Java System Access Manager

Managing policies through Sun Java System Access Manager gives you control of the same privileges available with the access control file method; however it additionally allows more fine-tuned control over various features, such as: the ability to receive alerts, send polls, receive polls, etc. For a complete list, please refer to table Table 6-4 on page 113. Furthermore, managing policies using Sun Java System Access Manager gives you finer-tuned control over privileges.

Two types of policies exist: Instant Messaging policies and Presence policies. The Instant Messaging policies govern general Instant Messaging features, such as the ability to send or receive alerts; the ability to manage public conferences and news channels; and the ability to send files. Presence policies govern the control end users have over changing their online status, and in allowing or preventing others from seeing their online or presence information.

Managing Policies: The Method to Use

When choosing which method to use to manage policies, it is also necessary to choose where they will be stored. You select the method for managing policies by editing the `iim.conf` file and setting the `iim.policy.modules` parameter to either `identity` for the Access Manager method or `iim_ldap` for the access control file method, which is also the default method.

If you will use an LDAP-only deployment—therefore, you will not be using Sun Java System Access Manager—you must use the access control file method. If you are using Sun Java System Access Manager with the Instant Messaging server, and you have installed the Instant Messaging and Presence services components, you can use either policy management method. Managing policies using Sun Java System Access Manager is a more comprehensive method. One advantage of this method is that it allows you to store all end-user information in the directory.

The specific steps for setting which method you want to use to manage policies are as follows:

1. Change directories to the directory that contains the `iim.conf` file.
2. Open the `iim.conf` file using an editor of your choice.
3. Edit the `iim.policy.modules` parameter by setting it to one of the following:
    - `iim_ldap` (the access control file method)
    - `identity` (the Access Manager method)
4. Edit the `iim.userprops.store` parameter and set it to either:
    - ldap (to store user properties in LDAP)
    - file (default, to store user properties in files)
5. Save your changes.
6. Refresh the configuration.

Policy Configuration Parameters

Table 6-1 lists and describes the parameters available in the `iim.conf` file that relate to the increased role that Sun Java System Access Manager can play in Instant Messaging deployments:

Table 6-1 Parameters Related to Access Manager in iim.conf

Parameter Name	Use	Values
`iim.policy.modules`	Indicates if Sun Java System Access Manager is used for policy storage	iim_ldap (default) identity
`iim.userprops.store`	Indicates if the user properties are in user properties file or from LDAP	file (default) ldap

NOTE Currently the `iim.userprops.store` parameter is only significant when the service definitions for the Presence and Instant Messaging services have been installed.

Managing Policies Using Access Control Files

By editing access control files you control the following end-user privileges:

- To access the presence status of the other end users
- To send alerts to other end users
- To save properties on the server
- To create new conference rooms
- To create new news channels

By default, end users are provided the privileges to access the presence status of other end users, send alerts to end users, and save properties to the server. In most of the deployments, the default values need not be changed.

NOTE Although certain privileges can be set globally, the administrator can also define exceptions for these privileges. For example, the administrator can deny certain default privileges to select end users or groups.

The location of the access control files is *im_cfg_base*/`acls`. Where *im_cfg_base* is the configuration directory. See Table 3-1 on page 37 for information about the default location of the configuration directory.

Table 6-2 lists the global access control files for Instant Messaging and the privileges these files provide end users.

Table 6-2 Access Control Files

ACL File	Privileges
`sysSaveUserSettings.acl`	Defines who can and cannot change their own preferences. Users who do not have this privilege cannot add contacts, create conferences, etc.
`sysTopicsAdd.acl`	Defines who can and cannot create News channels.
`sysRoomsAdd.acl`	Defines who can and cannot create Conference rooms.
`sysSendAlerts.acl`	Defines who can and cannot send alerts.
`sysWatch.acl`	Defines who can and cannot watch changes of other end users. The Instant Messenger window is displayed for end users who do not have this privilege allowing 'conference and news channel subscription and non-subscription' only.
`sysAdmin.acl`	Reserved for administrators only.This file sets administrative privileges to all Instant Messaging features for all end users. This privilege overrides all the other privileges and gives the administrator MANAGE access to all conference rooms and news channels as well as to end user presence information, settings, and properties.

Access Control File Format

The access control file contains a series of entries that define the privileges. Each entry starts with a tag as follows:

- `d:` - default
- `u:` - user
- `g:` - group

NOTE The `d:` tag must be the last entry in an access control file. The server ignores all entries after a `d:` tag. If the `d:` tag is `true`, then all other lines are ignored. You cannot set the `d:tag` as `true` in an access control file and selectively disallow end users that privilege.

The tag is followed by a colon (`:`). In case of the default tag it is followed by `true` or `false`.

End-user and group tags are followed by the end-user or group name.

Multiple end users and groups are specified by having multiple end users (`u`) and groups (`g`) in lines.

If default is set to `true`, all other entries in the file are redundant. If default is set to `false`, only the end users and groups specified in the file will have that particular privilege.

The following are the default `d:` tag entries in the ACL files for a new installation:

- `sysAdmin.acl` - Contains `d:false`
- `sysTopicsAdd.acl` - Contains `d:false`
- `sysRoomsAdd.acl` - Contains `d:false`
- `sysSaveUserSettings.acl` - Contains `d:true`
- `sysSendAlerts.acl` - Contains `d:true`
- `sysWatch.acl` - Contains `d:true`

NOTE The format and also the existence of all the access control files might change in future releases of the product.

Access Control File Examples

This section shows a sample access control file that shows privileges set for the sysTopicsAdd.acl file.

sysTopicsAdd.acl File

In the following example, the default d: tag entry for sysTopicsAdd.acl file is false. So the Add and the Delete news channels privileges are available to the end users and groups that appear before the default, namely user1, user2, and the sales group.

```
# Example sysTopicsAdd.acl file
u:user1
u:user2
g:cn=sales,ou=groups,o=siroe
d:False
```

Changing End User Privileges

➤ To Change End-user Privileges

1. Change to the config/acls directory. For example, on Solaris:

   ```
   cd /etc/opt/SUNWiim/default/config/acls
   ```

2. Edit the appropriate access control file. For example:

   ```
   vi sysTopicsAdd.acl
   ```

3. Save the changes.
4. End users need to refresh the Instant Messenger window to see the changes.

Managing Policies using Sun Java System Access Manager

The Instant Messaging and Presence services in Sun Java System Access Manager provide another way to control end user and administrator privileges. Each service has three types of attributes: dynamic, user, and policy. A policy attribute is the type of attribute used to set privileges.

Policy attributes become a part of the rules when rules are added to a policy created in Access Manager to allow or deny administrator and end-user involvement in various Instant Messaging features, such as receiving poll messages from others.

When Instant Messaging server is installed with Sun Java System Access Manager, several example policies and roles are created. See the *Sun Java System Access Manager Getting Started Guide* and the *Sun Java System Access Manager Administration Guide* for more information about policies and roles.

Furthermore, if the example policies are not sufficient, you can create new policies and assign those policies to a role, group, organization, or end user as needed to match your site's needs.

When the Instant Messaging service or the Presence service are assigned to end users, they receive the dynamic and user attributes applied to them. The dynamic attributes can be assigned to a Sun Java System Access Manager configured role or organization.

When a role is assigned to an end user or an end user is created in an organization, the dynamic attributes then become a characteristic of the end user. The user attributes are assigned directly to each end user. They are not inherited from a role or an organization and, typically, are different for each end user.

When end users log on, they get all the attributes that are applicable to them depending upon which roles are assigned to them and how the policies are applied.

Dynamic, user or policy attributes are associated with end users after assigning the Presence and Instant Messaging Services to these end users.

Instant Messaging Service Attributes

Table 6-3 lists the policy, dynamic, and user attributes that each service has.

Table 6-3 Access Manager Attributes for Instant Messaging

Service	Policy Attribute	Dynamic Attributes	User Attributes
sunIM	sunIMAllowChat	sunIMProperties	sunIMUserProperties
	sunIMAllowChatInvite	sunIMRoster	sunIMUserRoster
	sunIMAllowForumAccess	sunIMConferenceRoster	sunIMUserConferenceRoster
	sunIMAllowForumManage	sunIMNewsRoster	sunIMUserNewsRoster
	sunIMAllowForumModerate	sunIMPrivateSettings	sunIMUserPrivateSettings
	sunIMAllowAlertsAccess		
	sunIMAllowAlertsSend		
	sunIMAllowNewsAccess		
	sunIMAllowNewsManage		
	sunIMAllowFileTransfer		
	sunIMAllowContactListManage		
	sunIMAllowUserSettings		
	sunIMAllowPollingAccess		
	sunIMAllowPollingSend		
sunPresence	sunPresenceAllowAccess	sunPresenceDevices	sunPresenceEntityDevices
	sunPresenceAllowPublish	sunPresencePrivacy	sunPresenceUserPrivacy
	sunPresenceAllowManage		

For each attribute in the preceding table, a corresponding label appears in the Access Manager admin console. The two following tables list each attribute with its corresponding label and a brief description. Table 6-4 lists and describes the policy attributes and Table 6-5 lists and describes the dynamic and user attributes.

Table 6-4 Access Manager Policy Attributes for Instant Messaging

Policy Attribute	Admin Console Label	Attribute Description
`sunIMAllowChat`	Ability to Chat	End users can be invited to join chat room and access normal chat functionality
`sunIMAllowChatInvite`	Ability to Invite others to Chat	End users can invite others to chat

Table 6-4 Access Manager Policy Attributes for Instant Messaging *(Continued)*

Policy Attribute	Admin Console Label	Attribute Description
`sunIMAllowForumAccess`	Ability to Join Conference Rooms	A conference tab shows up in Instant Messenger, allowing end users to join conference rooms
`sunIMAllowForumManage`	Ability to Manage Conference Rooms	End users are able to create, delete, and manage conference rooms
`sunIMAllowForumModerate`	Ability to Moderate Conference Rooms	End users can be conference moderators
`sunIMAllowAlertsAccess`	Ability to Receive Alerts	End users can receive alerts from others
`sunIMAllowAlertsSend`	Ability to Send Alerts	End users can send alerts to others
`sunIMAllowNewsAccess`	Ability to Read News	A News button is displayed in Instant Messenger that enables end users to list news channels in order to receive and send news messages
`sunIMAllowNewsManage`	Ability to Manage News Channels	End users can manage news channels and create, delete, and assign privileges to news channels
`sunIMAllowFileTransfer`	Ability to Exchange Files	End users can add attachments to alert, chat, and news messages
`sunIMAllowContactListManage`	Ability to Manage one’s Contact List	End users can manage their own contact lists; they can add and delete users or groups to and from the list; they can rename the folder in their contact list
`sunIMAllowUserSettings`	Ability to Manage Messenger	A Settings button is displayed in the Instant Messenger that enables end users to change their own Instant Messenger settings
`sunIMAllowPollingAccess`	Ability to Receive Polls	End users can receive poll messages from others, and they can respond to polls
`sunIMAllowPollingSend`	Ability to Send Polls	A Poll button is displayed in Instant Messenger that enables end users to send poll messages to others and to receive the responses
`sunPresenceAllowAccess`	Ability to Access other’s Presence	End users can watch the presence status of others. The contact list, in addition to showing the contact, reflects contacts’ presence status changes by changing the status icon

Table 6-4 Access Manager Policy Attributes for Instant Messaging *(Continued)*

Policy Attribute	Admin Console Label	Attribute Description
`sunPresenceAllowPublish`	Ability to Publish Presence	End users can click to select their status (online, offline, busy, etc.) for others to watch
`sunPresenceAllowManage`	Ability to Manage Presence Access	An Access tab is displayed in the Settings of the Instant Messenger; end users can set up their own default presence access, presence permitted, or presence denied list

Modifying Attributes Directly

An end user can log into Sun Java System Access Manager admin console and view the values of attributes in the Instant Messaging and Presence service attributes. If the attributes have been defined as modifiable, end users can alter them. However, by default no attributes in the Instant Messaging service are modifiable, nor is it recommended that end users be allowed to modify them. However, from the standpoint of system administration, manipulating attributes directly can be useful.

For example, since roles do not affect some system attributes, such as setting conference subscriptions, system administrators might want to modify the values of these attributes by copying them from another end user (such as from a conference roster) or modifying them directly. These attributes are listed in Table 6-5 on page 116.

In reference to table Table 6-5,user attributes can be set by end users through the Sun Java System Access Manager admin console. Dynamic attributes are set by the administrator. A value set for a dynamic attribute overrides or is combined with the corresponding user attribute value.

The nature of corresponding dynamic and user attributes influences how conflicting and complementing information is resolved. For example, Conference Subscriptions from two sources (dynamic and user) complement each other; therefore, the subscriptions are merged. Neither attribute overrides the other.

Table 6-5 Access Manager User and Dynamic Attributes for Instant Messaging

Admin Console Label	User Attribute	Dynamic Attribute	Attribute Description	Conflict Resolution
Messenger Settings	`sunIMUserProperties`	`sunIMProperties`	Contains all the properties for Instant Messenger and corresponds to the `user.properties` file in the file-based user properties storage	Merge-however, if a particular property has a value from both the user and dynamic attribute, the dynamic attribute overrides.
Subscriptions	`sunIMUserRoster`	`sunIMRoster`	Contains subscription information (user contact list roster)	Merge- if a Jabber identifier is present in both the user and dynamic attribute, then the nickname will be taken from the user attribute, the group will be a union of all groups from both user and dynamic attributes, the subscription value will be the highest value from the user and dynamic value.
Conference Subscriptions	`sunIMUserConferenceRoster`	`sunIMConferenceRoster`	Contains conference room subscription information	Merge-dynamic and user subscriptions are merged, and duplicates are removed.
News Channel Subscriptions	`sunIMUserNewsRoster`	`sunIMNewsRoster`	Contains news channel subscription information	Merge-dynamic and user subscriptions are merged and duplicates are removed.
Presence Agents	`sunPresenceEntityDevices`	`sunPresenceDevices`	Not used in this release (for future use)	The dynamic information is used.

Table 6-5 Access Manager User and Dynamic Attributes for Instant Messaging *(Continued)*

Admin Console Label	User Attribute	Dynamic Attribute	Attribute Description	Conflict Resolution
Privacy	`sunPresenceUserPrivacy`	`sunPresencePrivacy`	Corresponds to the privacy setting in Instant Messenger	Merge - the dynamic value is taken if there is a conflict.
Instant Messenger Preferences	`sunIMUserPrivateSettings`	`sunIMPrivateSettings`	Store private preferences here that are not stored in Messenger Settings.	Merge

Predefined Examples of Instant Messaging and Presence Policies

Table 6-6 lists and describes the seven example policies and roles that are created in Sun Java System Access Manager when the Instant Messaging service component is installed. You can add end users to different roles according to the access control you want to give them.

A typical site might want to assign the role IM Regular User (a role that receives the default Instant Messaging and Presence access) to end users who simply use Instant Messenger, but have no responsibilities in administering Instant Messaging policies. The same site might assign the role of IM Administrator (a role associated with the ability to administer Instant Messaging and Presence services) to particular end users with full responsibilities in administering Instant Messaging policies. Table 6-7 lists the default assignment of privileges amongst the policy attributes. If an action is not selected in a rule, the values *allow* and *deny* are not relevant as the policy then does not affect that attribute.

Table 6-6 Default Policies and Roles for Sun Java System Access Manager

Policy	Role the Policy Applies to	Service the Policy Applies to	Policy Description
Default Instant Messaging and presence access	IM Regular User	sunIM, sunPresence	The default access that a regular Instant Messaging end user should have.
Ability to administer Instant Messaging and Presence Service	IM Administrator	sunIM, sunPresence	The access that an Instant Messaging Administrator has, which is access to all Instant Messaging features.

Table 6-6 Default Policies and Roles for Sun Java System Access Manager *(Continued)*

Policy	Role the Policy Applies to	Service the Policy Applies to	Policy Description
Ability to manage Instant Messaging news channels	IM News Administrator	sunIM	End users can manage news channels by creating, deleting, etc.
Ability to manage Instant Messaging conference rooms	IM Conference Rooms Administrator	sunIM	End users can manage conference rooms by creating, deleting, etc.
Ability to change own Instant Messaging user settings	IM Allow User Settings Role	sunIM	End users can edit settings by clicking the Setting button in the Instant Messenger.
Ability to send Instant Messaging alerts	IM Allow Send Alerts Role	sunIM	End users can send alerts in Instant Messenger.
Ability to watch changes on other Instant Messaging end users	IM Allow Watch Changes Role	sunIM	End users can access the presence status of other Instant Messaging end users.

Table 6-7 Default Policy Assignments

	Policy						
Attribute	**Default access**	**Can administer Instant Messaging and Presence Service**	**Can manage news channels**	**Can manage conference rooms**	**Can change own end-user settings**	**Can send alerts**	**Can watch changes to other users**
sunIMAllowChat	allow	allow					
sunIMAllowChatInvite	allow	allow					
sunIMAllowForumAccess	allow	allow		allow			
sunIMAllowForumManage	deny	allow		allow			
sunIMAllowForumModerate	deny	allow		allow			
sunIMAllowAlertsAccess	allow	allow				allow	
sunIMAllowAlertsSend	allow	allow				allow	
sunIMAllowNewsAccess	allow	allow	allow				
sunIMAllowNewsManage	deny	allow	allow				
sunIMAllowFileTransfer	allow	allow					
sunIMAllowContactListManage	allow	allow					

Table 6-7 Default Policy Assignments *(Continued)*

	Policy						
Attribute	**Default access**	**Can administer Instant Messaging and Presence Service**	**Can manage news channels**	**Can manage conference rooms**	**Can change own end-user settings**	**Can send alerts**	**Can watch changes to other users**
sunIMAllowUserSettings	allow	allow			allow		
sunIMAllowPollingAccess	allow	allow					
sunIMAllowPollingSend	allow	allow					
sunPresenceAllowManage	allow	allow					
sunPresenceAllowAccess	allow	allow					allow
sunPresenceAllowPublish	allow	allow					

Creating New Instant Messaging Policies

You can create new policies to fit the specific needs of your site.

➤ To Create a New Policy

1. Log on to the Access Manager admin console at http://hostname:port/amconsole, for example http://imserver.company22.example.com:80/amconsole
2. With the Identity Management tab selected, select Policies in the View drop down list in the navigation pane (the lower-left frame).
3. Click New to bring up the New Policy page in the data pane (the lower-right frame).
4. Select Normal for the Type of Policy.
5. Enter a policy description in the Name field, such as Ability to Perform IM Task.
6. Click Create to make the name of the new policy appear on the policy list in the navigation pane and to make the page in the data pane change to the Edit page for your new policy.
7. In the Edit page, select Rules in the View drop down list to bring up the Rule Name Service Resource panel inside the Edit page.

8. Click Add to bring up the Add Rule page.

9. Select the Service that applies, either Instant Messaging Service or Presence Service.

   Each service enables you to allow or deny end users the ability to perform specific actions. For example, Ability to Chat is an action specific to the Instant Messaging service while Ability to Access other's Presence is an action specific to the Presence service.

10. Enter a description for a rule in the Rule Name field, such as Rule 1.

11. Enter the appropriate Resource Name (`IMResource` or `PresenceResource`):

    ❍ `IMResource` for Instant Messaging Service

    ❍ `PresenceResource` for Presence Service

12. Select the Actions that you want to apply.

13. Select the Value for each action: Allow or Deny.

14. Click Create to display this proposed rule in the list of saved rules for that policy.

15. Click Save to make this proposed rule a saved rule.

16. Repeat steps 8-15 for any additional rules that you want to apply to that policy. For each new rule, click Save to save the changes to the policy.

Assigning Policies to a Role, Group, Organization, or User

You can assign policies—the default policies for Instant Messaging or Instant Messaging policies that might have been created after Instant Messaging was installed—to a role, group, organization, or user.

➤ To Assign a Policy

1. Log on to the Access Manager admin console at http://hostname:port/amconsole, for example http://imserver.company22.example.com:80/amconsole
2. With the Identity Management tab selected, select Policies in the View drop down list in the navigation pane (the lower-left frame).
3. Click the arrow next to the name of the policy you want to assign in order to bring up the Edit page for that policy in the data pane (the lower-right frame).
4. In the Edit page, select Subjects in the View drop down list.
5. Click Add to bring up the Add Subject page, which lists the possible subject types:
   - Access Manager Roles
   - LDAP Groups
   - LDAP Roles
   - LDAP Users
   - Organization
6. Select the subject type that matches the policy, such as Organization.
7. Click Next
8. In the Name field, enter a description of the subject.
9. If desired, select the Exclusive check box.

   The Exclusive check box is not selected as the default setting, which means that the policy applies to all members of the subject.

   Selecting the Exclusive check box applies the policy to everyone who is not a member of the subject.

10. In the Available field, search for entries that you want to add to your subject.
    a. Type a search for the entries you want to search for. The default search is *, which displays all the subjects for that subject type.
    b. Click search.
    c. Highlight entries in the Available text box that you want to add to the Selected text box.
    d. Click Add or Add All, whichever applies.
    e. Repeat steps a-d until you have added all the names you want to the Selected text box.
11. Click Create to display this proposed subject in the list of saved subjects for that policy.
12. Click Save to make this proposed subject a saved subject.
13. Repeat steps 5-12 for any additional subjects that you want to add to the policy. For each new subject, click Save to save the changes to the policy.

Creating New Suborganizations Using Access Manager

The ability to create suborganizations using Sun Java System Access Manager enables organizationally separate populations to be created within the Instant Messaging server. Each suborganization can be mapped to a different DNS domain. End users in one suborganization are completely isolated from those in another. The following describes minimal steps to create a new suborganization for Instant Messaging.

➤ To Create a New Suborganization

1. Log on to the Access Manager admin console at `http://hostname:port/amconsole`, for example `http://imserver.company22.example.com:80/amconsole`
2. Create a new organization:
    a. With the Identity Management tab selected, select Organizations in the View drop down list in the navigation pane (the lower-left frame).
    b. Click New to bring up the New Organization page in the data pane (the lower-right frame).

c. Enter the following in the appropriate fields:

   - A suborganization name, such as `sub1`
   - A domain name, such as `sub1.company22.example.com`,

d. Click Create.

3. Register services for the newly created suborganization.

   a. Click the name for the new suborganization, such as `sub1`, in the navigation pane (Be certain to click the name, not the property arrow at the right.).

   b. Select Services from the View drop down list in the navigation pane

   c. Click Register to bring up the Register Services page in the data pane.

   d. Select the following services under the Authentication heading:

      - Core
      - LDAP

   e. Select the following services under the Instant Messaging Configuration heading:

      - Instant Messaging Service
      - Presence Service

   f. Click Register to bring up the newly selected services for this suborganization in the navigation pane.

4. Create service templates for the newly selected services:

   a. In the navigation pane, click the property arrow for a service, starting with the Core service.

      The Create Service Template page appears in the data pane.

   b. In the data pane, click Create, which replaces the Create Service Template page with a page of template options for the service you have selected.

      You should click Create for each service even when you do not want to modify the template options.

c. Modify the options for the service template of each service as follows:

   I. **Core:** Generally, no options need to be modified; go to Step d.

   II. **LDAP:** Add the prefix of the new suborganization to the *DN to Start User Search* field. After adding the prefix, the final DN should be in this format:

   ```
   o=sub1,dc=company22,dc=example,dc=com
   ```

   Enter the LDAP password in the *Password for Root User Bind* and *Password for Root User Bind (confirm)* fields.

   Continue to Step d:

   III. **Instant Messaging Service:** Generally, no options need to be modified; go to Step d.

d. Click Save.

e. Repeat steps a through d until you have created service templates for each service.

Adding End Users to New Suborganizations

After new end users have been created in a suborganization they need to be assigned roles. Roles can be inherited from the parent organization as described in the following section.

➤ To Add End Users to a New Suborganization

1. Go to the parent organization and select Roles from the View drop down list. The specific steps are:

   a. Log on to the Access Manager admin console at `http://hostname:port/amconsole`, for example `http://imserver.company22.example.com:80/amconsole`

   b. With the Identity Management tab selected, select Roles in the View drop down list in the navigation pane (the lower-left frame).

2. Click on the property arrow to the right of the role you wish to assign in order to bring up a page for that role in the data pane (the lower-right frame).

3. Select Users from the View drop down list in the data pane.

4. Click Add to bring up the Add Users page.

5. Enter a matching pattern to identify users. For example, in the `UserId` field an asterisk,*, lists all users.
6. Click Filter to bring up the Select User page.
7. Display the parentage path in the Select User page:
    a. Select the *Show parentage path* check box.
    b. Click Refresh.
8. Select the users to be assigned to this role.
9. Click Submit.

Chapter 7

Managing The Instant Messaging Archive

This chapter explains how to manage and configure the Instant Messaging Archive.

This chapter contains the following sections:

Instant Messaging Archive Overview

The Instant Messaging archive captures instant messages and archives these messages in a Portal Server Search database. It enables the end user to query and retrieve these archived messages using the Search page on the Portal Server desktop.

Instant Messaging Archive contains the following components:

Archive and Retrieval Component. Portal Server Search component also known as Archive and Retrieval component is used to store the archived Instant Messages. The Instant Messaging archive data is indexed and can be assigned to categories and stored in the Portal Server Search database. For example, alert messages can be stored under the Alert category.

NOTE Storing data in separate categories helps in simplifying the search operation and enables quick retrieval of the archived data.

Instant Messaging Archive Search or Display Servlet. When the end user performs a search operation for documents matching certain criteria, the Portal Server Search fetches pages matching this criteria. These pages can be remote web pages or they could be Instant Messaging archive data also referred as Instant Messaging resource descriptors.

- For the remote web pages, the URL of the pages matching the criteria is listed in the Search Results List. When the end user clicks the URL of a web page in the Search Results List, the browser fetches this page from the remote web container.
- For the Instant Messaging Resource Descriptor, the archive data is stored in the Portal Server Search database and is not available as downloadable documents from the web container.

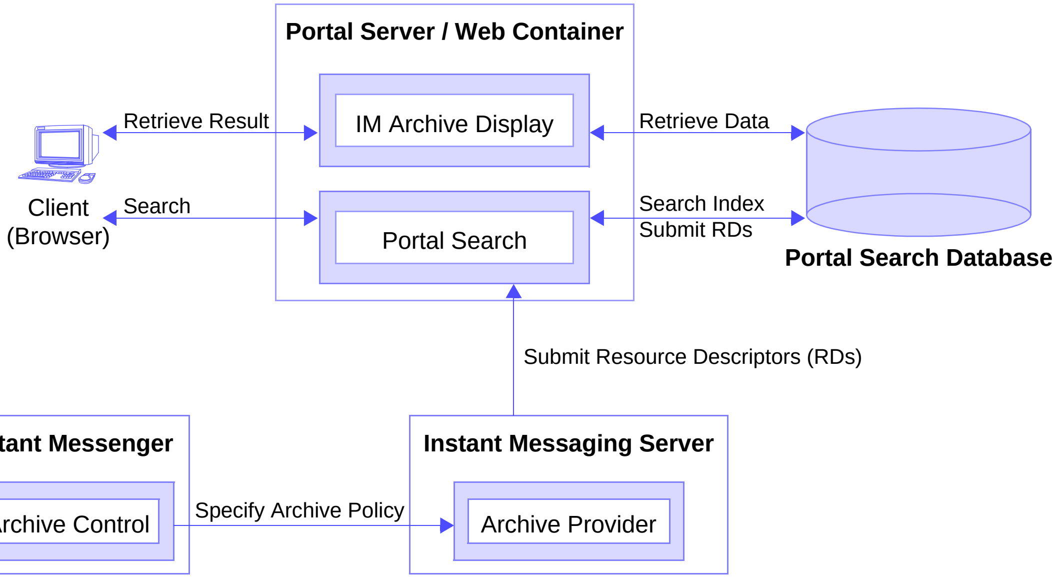
When the end user clicks the URL of the Instant Messaging resource descriptors to view the archive data, the Instant Messaging Archive Search or Display servlet is invoked. The Instant Messaging Archive Search servlet retrieves the information from the Portal Server Search database and generates a text or HTML response containing the Instant Messaging Archive data.

Instant Messaging Archive Provider. This component is invoked by the Instant Messaging server whenever Instant Messages are to be archived. The Instant Messaging Archive Provider builds the Summary Object Interchange Format (SOIF) complaint Resource Descriptors (RD) based on the data provided by the Instant Messaging server. It uses Portal Server Search APIs to send these Resource Descriptors to Portal Server Search database. It also maintains a buffer of the records to be submitted to the Portal Server Search database, to reduce the performance hit.

Instant Messenger Archive Control. Instant Messaging data can be archived automatically without any interaction from the end user. To control the archive functionality the end user needs to enable the Instant Messenger Archive Control component. This component allows the end user to set default archive options, such as "archive all conferences", and change the default on a per-transaction basis. For example, the end user can choose not to archive the content of the conferences.

Figure 7-1 illustrates Instant Messaging Archive components.

Figure 7-1 Instant Messaging Archive Components



Archiving Instant Messages

All instant messages are divided into the following categories for the purpose of archiving:

Chat. All the messages in the private conference rooms.

Conference. All the messages in the public conference rooms.

Alerts. This category contains all the alert messages.

Poll. This category contains all poll messages.

News. This category contains all messages posted in the news channels.

The following are the features of Instant Messaging Archive Provider:

- It captures all the Instant Messaging traffic passing through the server.
- The archived data can be stored under separate categories on the Portal Server Search.
- Storing the data as separate categories helps in simplifying the search and retrieval of the archived data.
- The search can be performed using the Portal Server desktop.
- The security feature of Portal Server Search can be used to provide an access control list. The archive provider provides security features by which only a set of admin users can be allowed to access the archived data.
- The data can be managed using the Portal Server Search database management tools.

Enabling the Archive Provider

➤ To Enable Archive Provider in Instant Messaging

1. Change to the `config` directory. For example, on Solaris:

```
cd /etc/opt/SUNWiim/default/config
```

2. Open the `iim.conf` file.

For example:

```
vi iim.conf
```

3. Add the following line to the `iim.conf` file:

   For the default archive provider, add the following line:

   ```
   iim_server.msg_archive = true
   ```

   For a custom archive provider, add the following line:

   `iim_server.msg_archive.provider =` *provider_name*

   To use the Portal Server Search based archive, replace the *provider_ name* with the following:

   ```
   com.iplanet.im.server.IMPSArchive
   ```

4. Save the file.

5. Refresh the Instant Messaging server configuration. To refresh type:

   ```
   imadmin refresh
   ```

Instant Messaging server provides the APIs and SPIs that can be used to write custom archive providers. For more information on Instant Messaging APIs, see "Instant Messaging APIs" on page 185.

Configuring the Archive Provider

The archive provider stores the archived messages as resource descriptors (RD) in the Portal Server Search database. The archive provider uses the following fields of the Portal Server Search schema:

Title. This field contains the names of the public conference rooms for Conference category, names of the participants in a chat session for the Chat category, subject of the Alert messages and the names of the News Channels for alerts and news categories. The title field will contain "`Poll from` *Sender*" for the poll category, where *Sender* represents the display name of the sender of the poll.

Keyword. For conference and chat categories, this field will contain a list of all the participants in the conference room. For a public conference room, it will also contain the name of the conference room. For the Alert category, it will contain the display names of the sender and the recipients. For the News category, it will contain the name of the channel. For the Polls category, it will contain the list of sender and recipients. For all categories, in addition to the above values this field also contains a unique ID for the categories.

Table 7-1 shows the unique ID and gives a description for each category in the archive provider.

Table 7-1 Unique ID for each category and their description

Category	Unique ID
Conference	`RoomName-StartTime`
Chat	Where: `RoomName` - Name of the public or private conference room `StartTime` - Is the timestamp of the creation of RD
Alert	`Alert-messageID` Where: `messageID` - Message Id of the message which will be archived. Message Id has importance when the RD contains only one message. For example, News message and Alert message.
Poll	`Poll-pollID`
News	`TopicName-messageID`

ReadACL. For the Conference and News categories, the value for this field is taken from the access control files of the respective conference rooms and news channels. For the Chat category, this field contains the DN of the participants. For the Alert category, this field contains the sender's DN and the recipient's DN. For the Poll category, the archiver will provide a new access control file.

The search access to the RDs is controlled by the value in the ReadACL field. If the document level security is enabled, the end user has access to the search results only if the ReadACL field has the end user's DN. If the Instant Messenger Archive control is enabled, for the chat messages, the end user DN added to the ReadACL field depends on the end-user selection.

Description. This field contains the archived message without the HTML formatting.

Full-Text.This field contains the HTML formatted archived messages.

Classification. This field contains the category of the archived message.

Archive Provider Configuration Parameters

Table 7-2 lists and describes the archive provider configuration parameters that can be added to the iim.conf file:

Table 7-2 Available archive Provider Parameters for iim.conf

Parameter	Default Value	Description
iim_arch.title.attr	Title	This parameter contains the name of the field equivalent to the Title field in the default schema of the Portal Server Search.
iim_arch.keyword.attr	Keyword	This parameter contains the name of the field equivalent to the Keyword field in the default schema of the Portal Server Search.
iim_arch.readacl.attr	ReadACL	This parameter contains the name of the field equivalent to the ReadACL field in the default schema of the Portal Server Search.
iim_arch.description.attr	Description	This parameter contains the name of the field equivalent to the Description field in the default schema of the Portal Server Search.
iim_arch.fulltext.attr	Full-Text	This parameter contains the name of the field equivalent to the Full-Text field in the default schema of the Portal Server Search.
iim_arch.category.attr	Category	This parameter contains the name of the field equivalent to the Category field in the default schema of the Portal Server Search.
iim_arch.readacl.admin	None	This parameter contains the administrator’s DN. Multiple values should be separated by “;”

Table 7-2 Available archive Provider Parameters for `iim.conf` *(Continued)*

Parameter	Default Value	Description
`iim_arch.readacl.adminonly`	`false`	This parameter will contain `true` or `false`. `true` - Only the administrator's DN specified by the parameter `iim_arch.readacl.admin` will be added to the `ReadACL` field overwriting the default behavior of the `ReadACL` field. `false` - The administrator's DN specified by the parameter `iim_arch.readacl.admin` will be added to the ReadACL field in addition to the default behaviour.

`iim_arch.categories`	`all`	This parameter contains a list of message types that can be archived.
		The value can be:
		`poll`
		`alert`
		`chat`
		`conference`
		`news`
		Multiple values can be specified separated by commas(“`,`”).

`iim_arch.categoryname`	None	If a category name is not assigned for any of the categories then the value of this parameter is taken as the category name.
`iim_arch.alert.categoryname`	None	This parameter contains the name of the category containing the archived alert messages. Note: It is not required to dedicate a category to alert messages.
`iim_arch.poll.categoryname`	None	This parameter contains the name of the category containing the archived poll messages. Note: It is not required to dedicate a category to poll messages.
`iim_arch.conference.categoryname`	None	This parameter contains the name of the category containing the archived conference messages. Note: It is not required to dedicate a category to conference messages.

Table 7-2 Available archive Provider Parameters for `iim.conf` *(Continued)*

Parameter	Default Value	Description
`iim_arch.chat.categoryname`	Name	This parameter contains the name of the category containing the archived chat messages. Note: It is not required to dedicate a category to chat messages.
`iim_arch.news.categoryname`	None	This parameter contains the name of the category containing the archived news messages. Note: It is not required to dedicate a category to news messages.
`iim_arch.conference.quiettime`	`5`	This parameter contains the maximum duration of silence between two consecutive messages in a room (both public and private) after which the RD expires and a new RD is created for archiving the message. The value is in minutes.
`iim_arch.poll.maxwaittime`	`15`	This parameter contains the (maximum) time for which poll data is buffered in the server. The value is in minutes.
`iim_arch.ignoreexplicitdeny`	`true`	This parameter will contain `true` or `false`. `true` - For Poll and Conference category the data with explicit deny access will not be archived. Each time when these messages are not archived this information will be logged into the `xmppd.log` file. `false` - For Poll and Conference category the data with explicit deny access will not be archived and the message will be added to the Portal Server Search database. Note: If you do not explicitly deny access to a room or a news channel then the default access is either READ or WRITE or MANAGE. Some end users can also be granted NONE access.

Table 7-2 Available archive Provider Parameters for `iim.conf` *(Continued)*

Parameter	Default Value	Description
`iim_arch.portal.search`	`None`	The value of the this parameter should be the URL of the Portal Server Search servlet. For example: `http://www.example.com/portal/search` If this parameter is not present then the Archive Provider determines the value of the Portal Server Search URL based on the `AMConfig.properties` file present on the system.
`iim_arch.portal.admindn`	`None`	The value of this parameter should be the dn of the admin user. For example: `uid=amadmin,ou=People,o=internet` This parameter is required when the Document level Security in the Portal Search Server is on.
`iim_arch.portal.adminpassword`	`None`	The value of this parameter should be the password of the admin user as specified by the `iim_arch.portal.admindn` parameter. This parameter is required when the Document level Security in the Portal Search Server is on.
`iim_arch.portal.search.database`	`None`	The value of this parameter should be the name of the database where the Instant Messaging server stores archived messages. If this parameter is not defined then all messages are stored in the default database of Sun Java System Portal Server Search.

Storing Archived Messages in a Non-default Portal Server Search Database

➤ To Store Archived Messages in a Non-default Database

1. Modify the `iim.conf` file.

   a. Change to the `config` directory. For example, on Solaris:

      `cd /etc/opt/SUNWiim/default/config`

   b. Open the `iim.conf` file using an editor of your choice.

      For example, you could type:

      `vi iim.conf`

   c. For the default archive provider, add the following line:

      `iim_arch.portal.search.database =` *database-name*

      where ***database-name*** is the name of your non default database.

   d. Save the file.

2. Modify the Portal Server Search Channel.

   Change the Portal Server Search Channel to add an option for searching the data in another database. See the *Sun Java System Portal Server Desktop Customization Guide* for more information.

3. Modify the `IMArchiveDisplay.jsp` file:

   a. Change to the following directory:

   `/etc/opt/SUNWps/desktop/default/IMProvider/`

   b. Create a back up of the `IMArchiveDisplay.jsp` file.

   c. Edit the `IMArchiveDisplay.jsp` file with an editor of your choice. For example, you could type the following:

      `vi IMArchiveDisplay.jsp`

   d. Search through the `IMArchiveDisplay.jsp` file and locate the following two lines of code:

Code Example 7-1 Search Code from `IMArchiveDisplay.jsp` File, Before Editing

```
<search:setQuery query = "<%= scope %>"/>
<search:setRDMType rdmType = "rd-request"/>
```

e. Between the two lines of code shown in Code Example 7-1, add the following line of code:

`<search:setDatabase database = "`*database-name*`"/>`

After you add the new line of code, that section of code should look like Code Example 7-2:

Code Example 7-2 Search Code from `IMArchiveDisplay.jsp` File, After Editing

```
<search:setQuery query = "<%= scope %>"/>
 <search:setDatabase database = "database-name"/>
<search:setRDMType rdmType = "rd-request"/>
```

where ***database-name*** is the name of the non-default database.

Managing Archived Data in the Portal Server Search Database

NOTE These instructions are Solaris specific.

The Instant Messaging data is archived in the form of Resource Descriptors (RDs) in the Portal Server Search database. The individual entries in the Portal Server Search database are called resource descriptors (RDs). An RD is a specific set of information about a single resource. The fields of each RD are determined by the Portal Server Search database schema.

To manage the archived data, you need to manage the Resource Descriptors (RDs) in the Portal Server Search database. This section explains some of the frequently performed maintenance tasks on the Portal Server Search database.

For more information on managing data in the Portal Server Search database, see the *Sun Java System Portal Server Administration Guide*.

rdmgr Command

The `rdmgr` command is the main command used to work with the Search service. It gives the administrator two types of subcommands: one that is used to work with resource descriptors (RDs); and the other that is used for database maintenance. The `rdmgr` command is normally run in a search-enabled Portal Server instance directory.

➤ To Invoke the `rdmgr` Command

1. Change to the following directory:

   `cd /var/opt/SUNWps/https-`*servername*`/`

2. Type the following at the command-line:

   `run-cs-cli` *portal-server-install-dir*`/SUNWps/bin/rdmgr args`

   where *portal-server-install-dir* is the directory in which Portal Server is installed.

For more information on `rdmgr` command, see Command-Line Utilities in *Sun Java System Portal Server Administration Guide*.

Searching Resource Descriptors

Running `rdmgr` command with the argument value `-Q` generates a list of resource descriptors (RDs) that refines the search operation.

For example:

- To search for resource descriptors (RDs) containing the text `testing`, type:

  `run-cs-cli` *portal-server-install-dir*`/SUNWps/bin/rdmgr -Q testing`

- To search for resource descriptors (RDs) belonging to a particular category, type:

```
run-cs-cli portal-server-install-dir/SUNWps/bin/rdmgr -Q
"classification=Archive:Chat:January"
```

Deleting Resource Descriptors

The following are the examples for deleting resource descriptors (RDs) from the Portal Server Search database:

To delete all resource descriptors (RD) containing the text `testing`, type:

```
run-cs-cli portal-server-install-dir/SUNWps/bin/rdmgr -d -Q testing
```

To delete all resource descriptors (RD) from a category `Archive:Chat:January`, type:

```
run-cs-cli portal-server-install-dir/SUNWps/bin/rdmgr -d -Q
"classification=Archive:Chat:January"
```

Enabling Instant Messenger Archive Control

The Instant Messenger Archive Control component enables the end user to control the archived instant messages. This component allows the end user to search for the archived instant messages stored in the Portal Server Search database by clicking the Archive button in the Instant Messenger main window. It also enables the end user to set default archive options, such as “archive all conferences” in the Archive tab of the Instant Messenger. The Instant Messenger Archive Control feature is provided by two optional Instant Messenger modules.

The Instant Messenger Archive Control component can be enabled by setting the `archive_control` applet parameter in the applet descriptor file.

The applet descriptor files for the Instant Messaging LDAP deployment that need to be changed are:

- `im.jnlp, imssl.jnlp and jnlpLaunch.jsp (portal only)` for Java Web Start
- `im.html, imssl.html and pluginLaunch.jsp (portal only)` for Java Plugin

Changes for JNLP files and jnlpLaunch.jsp files:

If you are using Java Web Start to launch the Instant Messenger, perform the following steps to enable the Instant Messenger Archive Control feature in the Instant Messenger:

1. Go to the Instant Messenger documentation root directory to locate the `im.jnlp` and `imssl.jnlp` files

   The `jnlpLaunch.jsp` file can be found at:

   `/etc/opt/SUNWps/desktop/default/IMProvider`

2. Edit the `jnlp or jsp` file, and add or edit the following line:

```
<argument>archive_control=true</argument>
```

Changes for html applet pages and pluginLaunch.jsp files:

If you are using Java Plug-in to launch the Instant Messenger, perform the following steps to enable the Instant Messenger Archive Control feature in the Instant Messenger:

1. Go to the messenger documentation root directory to locate the `im.jnlp` and `imssl.jnlp` files

   The `jnlpLaunch.jsp` file can be found at:

   `/etc/opt/SUNWps/desktop/default/IMProvider`

2. Edit the `jnlp or jsp` file, and add or edit the following line:

```
<PARAM NAME="archive_control" VALUE="true" />
<EMBED archive_control=true;/>
```

NOTE Instant Messenger Archive Control should not be enabled if the value of `iim_server.msg_archive.auto` is set to true in the `iim.conf` file of the Instant Messaging Server, as end users' messenger settings will not have any effect.

Changing the Display of the Archived Data

The data that is archived is deployed using the `IMArchiveDisplay.jsp` file. The `IMArchiveDisplay.jsp` file is installed in the folder `/etc/opt/SUNWps/desktop/default/IMProvider` by default. The file can be modified to change the style and the resource strings of the archived data.

For example, to replace the default system message displayed when an end user joins the room "joe has joined the room" to "joe has entered the room;" perform the following:

1. Edit the `IMArchiveDisplay.jsp` file with an editor of your choice. For example, you could type the following:

   ```
   vi IMArchiveDisplay.jsp
   ```

2. Replace the code line in Code Example 7-3 with Code Example 7-4 in the file `IMArchiveDisplay.jsp`:

Code Example 7-3 Modifying the default system message.

```
....
ht.put("has_joined_the_room","<span class='user'> {0} </span>
<span class='headervalue'> has joined the room.</span>");
....
```

Code Example 7-4 After replacing the default system message.

```
....
ht.put("has_joined_the_room","<span class='user'> {0} </span>
<span class='headervalue'> has entered the room.</span>");
....
```

Similarly, the resource strings for the other keys and the style for displaying the key information can also be modified.

If the attribute name of Title and Full-Text in the default schema of the Portal Server Search is changed, then these changes should also be reflected in the `IMArchiveDisplay.jsp` file.

Sample Deployment Scenario for Archive Provider

This sample deployment scenario explains how to archive the related Instant Messaging data collectively.

➤ To Archive the Related Instant Messaging Data Collectively

Create separate categories for each type of data. For example, in the Archive category where all the archived Instant Messaging data are stored, create a subcategory Chat for storing chat messages. You can also create subcategories for archiving data based on time. For example, to archive chat data for the month of December 2002 the subcategory will be:

```
Archive:Chat:2002:12
```

➤ To Archive All Chat Data Based on Time

1. Change to the `config` directory. For example, on Solaris type:

   ```
   cd /etc/opt/SUNWiim/default/config
   ```

2. Edit the `iim.conf` file. For example:

   ```
   vi iim.conf
   ```

3. Add the following value for the parameter `iim_arch.chat.categoryname`:

   ```
   iim_arch.chat.categoryname = Archive:Chat:%Y:%M
   ```

   The archive provider automatically assigns the current year for %Y and current month for %M. These values are taken from the system date and time.

➤ To Archive and Back up Chat Data for the Month of December 2002 to the Subcategory

1. Type the following:

   ```
   rdmgr -Q "classification=Archive:Chat:2002:12" > archive.soif
   ```

2. Store the `.soif` file to your backup system.

➤ To Remove Archived Chat Data for the Month of December 2002 from the Portal Server Search Database

- Type the following:

  ```
  rdmgr -d "classification=Archive:Chat:2002:12"
  ```

Chapter 8

Troubleshooting and Monitoring Instant Messaging

This chapter lists the common problems that might occur during installation and deployment of Instant Messaging and provides an overview of the monitoring agent. The log information generated by the various system components on their operation can be extremely useful when trying to isolate or troubleshoot a problem. In addition, you can use the monitoring agent to monitor the general health of Instant Messaging processes to help prevent problems before they occur, asses usage levels to help you scale your deployment, and to limit downtime. This chapter contains information in the following sections:

- Obtaining Instant Messenger Runtime Information
- Obtaining Instant Messenger Logs
- Problems and Solutions
- Troubleshooting Instant Messaging and LDAP
- Monitoring Instant Messaging
- Managing the Watchdog Process

For details and more information on managing server, multiplexor, watchdog, Calendar agent, and client logging, and for default log file locations, see "Managing Logging" on page 50.

Obtaining Instant Messenger Runtime Information

You can obtain information about a client system from the Instant Messenger client.

➤ **To Obtain Instant Messenger Runtime Information from the About Dialog**

1. In Instant Messenger, select About from the Help menu.

   The About dialog box appears.

2. Select the Details tab.

   The Details tab contains information about the client system that you can use when troubleshooting problems.

Obtaining Instant Messenger Logs

In order to obtain Instant Messenger logs, you first need to enable logging on the client host. See "Administering Client Logging" on page 53 for instructions.

Problems and Solutions

Listed below are some problems and their possible causes and the clues for troubleshooting these problems:

- Single sign-on does not work
- The Messenger client does not load or start
- Connection refused or timed out
- Authentication errors
- IM channel display error
- Instant Messaging content is not archived
- Server-to-server communication fails to start
- Catastrophic Installation Failure Leaves Server in an Inconsistent State

Single sign-on does not work

If you are using SSO with Sun Java System Access Manager, the Access Manager server and Instant Messaging server must be configured to use the same web container.

The Messenger client does not load or start

The following are the possible causes for this problem:

- Wrong codebase in the applet page.
- Application/x-java-jnlp-file MIME type not defined in the web container configuration.
- Plug-in of Java Web Start not installed or functional.
- No compatible Java version available.
- Security exception, cannot verify signature of `.jar` files.

Where to get the necessary information:

- In the Java Web Start or plug-in errors (exception stack trace, launch page.)
- In the applet page source on the browser.

Connection refused or timed out

The following are the possible causes for this problem:

- Either the Instant Messaging server or the multiplexor is not running.
- Incorrect multiplexor host or port names used in the Applet descriptor file (`.jnlp` or `.html`.)
- Different SSL settings used between the Instant Messenger and the multiplexor.
- Client and server version mismatch.

Where to get diagnostic information:

- Instant Messaging server and multiplexor log files.
- Instant Messenger logs.

Authentication errors

The following are the possible causes for this problem:

- Problems while accessing the LDAP server, such as the LDAP server is not running, or a provisioning error has occurred (for example, a schema violation).
- End user not found.
- Invalid credentials.
- Invalid Access Manager session.

Where to get diagnostic information:

- Instant Messaging server, Identity authentication and LDAP log files.

In addition, in deployments using Sun Java System Access Manager, ensure that the user entries in your Directory contain the `iplanet-am-managed-person` objectclass. The Instant Messaging server uses this object class when it searches for valid users in an Access Manager deployment. For more information about this object class and how Access Manager uses it, refer to the Sun Java System Access Manager documentation.

IM channel display error

The following are the possible causes for this problem:

- Authentication error when the server cannot validate the session token.
- Instant Messaging channel is not configured properly. For example, incorrect Instant Messaging server host and/or port.
- Plug-in or Java Web Start is not installed or is not functional.
- End user not found and the Instant Messaging server cannot find the end user in the LDAP lookup.

Where to get diagnostic information:

- Instant Messaging server and Instant Messaging channel logs.

Instant Messaging content is not archived

The following are the possible causes for this problem:

- Content is actually archived but the end user has insufficient rights to access it.
- The content has not yet been committed to the Compass database.
- The archive provider has been disabled in the Instant Messaging server.

Where to get diagnostic information:

- In the Instant Messaging server and the archive log files.

Server-to-server communication fails to start

The following are the possible causes for this problem:

- Incorrect server identification.
- Mismatch in the SSL settings.

Where get diagnostic information:

The necessary information can be obtained from the two Instant Messaging server log files.

Catastrophic Installation Failure Leaves Server in an Inconsistent State

If a catastrophic error occurs while installing or uninstalling Instant Messaging, the system might be left in an inconsistent state. This results in both install and uninstall being unable to complete. In this circumstance, you must manually remove all the Instant Messaging components so that a fresh install can be attempted. The clean up procedure consists of removing packages and registry information.

1. Back up any information you might need in a future installation. See "Backing Up Instant Messaging Data" on page 69.

2. Manually edit the product registry information.

   For Solaris 9, issue the following command:

   `prodreg(1)`

   For all other systems:

   a. Edit `productregistry.xml` and remove all Instant Messaging XML elements from the file. By default, the `productregistry` XML file is stored in the following locations:

   ❍ Solaris: `/var/sadm/install/productregistry`

   ❍ Linux: `/var/tmp/productregistry`

   b. Remove the following packages or RPMs if they are still present:

      - SUNWiim
      - SUNWiimc
      - SUNWiimd
      - SUNWiimid
      - SUNWiimin
      - SUNWiimjd
      - SUNWiimm

Troubleshooting Instant Messaging and LDAP

The following LDAP issues might arise in a given deployment. Change the LDAP parameters in the `iim.conf` file accordingly.

Issue: Your directory does not permit anonymous bind. By default, Instant Messaging server performs an anonymous search of the LDAP directory. However, it is common for sites to prevent anonymous searches in their directory so that any random person cannot do a search and retrieve all the information.

Solution: If your site's directory is configured to prevent such anonymous searches, then Instant Messaging server needs to have a user ID and password it can use to bind and do searches. Use the `iim_ldap.usergroupbinddn` and `iim_ldap.usergroupbindcred` parameters to configure the necessary credentials.

Issue: Your site does not use the `uid` attribute for user authentication.

Solution: Use the `iim_ldap.loginfilter` parameter to set the attribute that is used by your directory for authentication. By default, this parameter is set to `uid`. Also, change any "filter" parameters that contains `uid` in its value.

Issue: You want to change how Instant Messenger displays contact names from the default.

Solution: The default attribute that Instant Messenger uses to display contact names is `cn`. Thus, contact names appear as `Frank Smith`, `Mary Jones`, and so on. Edit the `iim_ldap.userdisplay` and `iim_ldap.groupdisplay` parameters to a different attribute, such as `uid`.

Issue: Your directory is indexed to use wildcards.

Solution: Change the `iim_ldap.allowwildcardinuid` parameter to `True`. This parameter determines if the use of wildcards should be enabled for User IDs while doing a search. As most directory installations have User IDs indexed for exact searches only, the default value is `False`. Setting this value to `True` can impact performance unless User IDs are indexed for substring search.

Issue: Your directory uses nonstandard object/group classes.

Solution: Change the appropriate `iim_ldap.*` parameters, replacing `inetorgperson` and `groupofuniquenames` with your values.

Issue: Your directory does not use the `mail` attribute for email addresses. If so, Instant Messenger will not be able to forward instant messages to offline users as email messages.

Solution: By default, the `iim_ldap.user.mailattr` contains the value `mail`. Change this value to your site's value.

Issue: Your directory uses an attribute other than uid as the user id attribute

Solution: If the attribute “loginname” is used as the user id attribute:

```
iim_ldap.user.uidattr=loginname
```

Add the following index directives to the indexing rules in LDAP:

```
index login name eq
```

Monitoring Instant Messaging

Instant Messaging provides an agent to help you monitor activity. This agent is called the monitoring framework management agent, or mfwk agent. The mfwk agent is contained within the Common Agent Container (CAC). The CAC and the mfwk agent are installed when you installed Instant Messaging.

The mfwk agent makes XMPP module statistics available through the Java Monitoring and Management Console (JConsole). Table 8-1 describes the Instant Messaging services for which the agent exposes state and performance metrics.

Table 8-1 Instant Messaging Services Monitored by the mfwk Agent

Category	Services	Description
Authentication	auth	Authentications.
Discovery	disco	Discovery requests.
Messages	message	Information about alerts and one-on-one chat sessions between two clients such as the speed at which messages are sent.
Conference	muc-presence, muc-admin, and muc-message	Conference statistics such as leaving or joining a conference, conference administrative requests, and conference (group chat) messages relayed.
Presence	presence-subscribe, presence-unsubscribe, presence-probe, and presence-authorize	Presence informations such as updates and subscriptions.
Privacy	private-get, private-set, privacy-get, and privacy-set	Privacy details.
Roster	roster-get and roster-set	Roster information.
Search	search	Search statistics.

This section provides information about administering and troubleshooting the `mfwk` agent and JConsole, and how you can use the agent and JConsole to monitor Instant Messaging in the following sections:

- Administering the mfwk Agent
- Viewing Monitoring Data
- Troubleshooting the mfwk Agent

Administering the mfwk Agent

You use the `mfwkadm` command-line utility to administer the `mfwk` agent. For example, you can start, stop, and restart the agent, and set up new and view current performance monitoring jobs performed by the agent. Detailed instructions on using this utility are available in the `mfwkadm` man page. In addition, the agent runs inside the CAC. For information on the CAC, refer to the `cacaoadm` and `cacao` man pages. This section provides instructions for locating these man pages.

➤ To Access the mfwkadm and CAC man Pages

1. On the command line, check your `MANPATH` environment variable to see if the correct paths are already there.

   Table 8-2 lists the paths to the man pages.

   Table 8-2 mfwkadm and CAC man page paths

Component	Solaris Path	Linux Path
mfwkadm	`/opt/SUNWmfwk`	`/opt/sun/mfwk`
CAC	`/opt/SUNWcacao/man`	`/opt/sun/man`

2. If the correct path is not there, append the location of the `mfwkadm` utility and CAC man pages to your `MANPATH` environment variable. For example, on Solaris using C shell:

   ```
   setenv
   MANPATH=/usr/dt/man:/usr/man:/opt/SUNWmfwk:/opt/SUNWcacao/man
   ```

   On Linux, update `/etc/man.config` with the path to the man pages.

3. Verify that the man pages are accessible. For example:

   ```
   man mfwkadm
   ```

Viewing Monitoring Data

Use JConsole to view the information exposed by the `mfwk` agent. JConsole is a graphical console tool that enables you to monitor and manage Java applications and virtual machines in your network. Using JConsole, you can browse the server JVM and also observe the Instant Messaging services described in Table 8-1.

For more information about using JConsole, see the JConsole documentation at the following locations:

`http://java.sun.com/j2se/1.5.0/docs/tooldocs/share/jconsole.html`

`http://java.sun.com/j2se/1.5.0/docs/guide/management/jconsole.html`

➤ To View Instant Messaging Monitoring Information Using JConsole

1. Log in as root.

2. Set the `CLASSPATH` to include the location of the CAC, JConsole, and the JMX jar file.

NOTE The line should be entered as a single line.

On Solaris:

```
/opt/SUNWcacao/lib/cacao_cacao.jar:/opt/SUNWjdmk/5.1/lib/jmxremote_optional.jar:/usr/jdk/e
ntsys-j2se/lib/jconsole.jar
```

On Linux:

```
/opt/sun/cacao/share/lib/cacao_cacao.jar:/opt/sun/jdmk/5.1/lib/jmxremote_optional.jar:/us
r/jdk/entsys-j2se/lib/jconsole.jar
```

3. Run JConsole.

NOTE The command should be entered as a single line.

On Solaris:

```
/usr/jdk/entsys-j2se/bin/java sun.tools.jconsole.JConsole "service:jmx:cacao-jmxmp://loca
lhost;wellknown=true;username=root"
```

On Linux:

```
/usr/jdk/entsys-j2se/bin/java sun.tools.jconsole.JConsole "service:jmx:cacao-jmxmp://loca
lhost;wellknown=true;username=root"
```

4. On the MBeans tab, expand the XMPP tree.

   The service attributes and their values are listed within the tree. See Table 8-1 for a complete list of Instant Messaging services visible through JConsole.

Troubleshooting the mfwk Agent

If you are experiencing trouble using the `mfwk` agent to monitor Instant Messaging, ensure the following:

- The dependencies are installed. In particular, check to make sure the JDMK, CAC, and the `mfwk` agent are installed. To check, use the following commands.

  Solaris:

  ❍ For the `mfwk` agent: `pkginfo SUNWmfwk-agent`

  ❍ For the CAC: `pkginfo SUNWcacao`

  ❍ For JDMK, if the CAC is running, then the JDMK is installed. You can also check that the jar files are installed under `/opt/SUNWjdmk/`*version*`/lib` and `/opt/SUNWjdmk/`*version*`/bin`, where *version* is the version number of the JDMK, for example `5.1`.

  Linux:

  ❍ For the `mfwk` agent: `rpm -qi sun-mfwk-agent-1.0`

  ❍ For the CAC: `rpm -qi sun-cacao-1.0`

  ❍ For JDMK, if the CAC is running, then the JDMK is installed. You can also check that the jar files are installed under `/opt/sun/jdmk/`*version*`/lib` and `/opt/sun/jdmk/`*version*`/bin`, where *version* is the version number of the JDMK, for example `5.1`. In addition, you can use the following command:

    ```
    rpm -qi sun-jdmk-runtime-5.1
    ```

- The CAC is running. To get status on the CAC, use the following commands.

  Solaris: `/opt/SUNWcacao/bin/cacaoadm status`

  Linux: `/opt/sun/cacao/bin/cacaoadm status`

  If CAC is not running, start it as follows:

  Solaris: `/opt/SUNWcacao/bin/cacaoadm start`

  Linux: `/opt/sun/cacao/bin/cacaoadm start`

- The XMPP module is loaded within the CAC and running:

  ```
  /opt/SUNWcacao/bin/cacaoadm status com.sun.im.service.xmpp
  ```

- The `mfwk` agent is loaded within the CAC:

  Solaris: `/opt/SUNWcacao/bin/cacaoadm list-modules`

  Linux: `/opt/sun/cacao/bin/cacaoadm list-modules`

  The module name for the `mfwk` agent is `com.sun.mfwk.mfwk_module`.

- The `mfwk` agent is running. To get status on the `mfwk` agent, use the following commands.

  Solaris: `/opt/SUNWcacao/bin/cacaoadm status com.sun.mfwk.mfwk_module`

  Linux: `/opt/sun/cacao/bin/cacaoadm status com.sun.mfwk.mfwk_module`

  If the `mfwk` agent is not running, start it as follows:

  Solaris: `/opt/SUNWmfwk/bin/mfwkadm start`

  Linux: `/opt/sun/mfwk/bin/mfwkadm start`

Troubleshooting JConsole

If you cannot bring up JConsole, ensure the following:

- The path to JConsole is correctly entered in your `CLASSPATH`.
- You logged in as root before attempting to run JConsole.

Managing the Watchdog Process

The watchdog process monitors the server and multiplexor components and attempts to restart a component if it determines that the component is not running.

For the server, the watchdog determines whether the server is running by periodically attempting to make a connection, either directly to the server or through the multiplexor, based on the current configuration of the server. The watchdog tries to poll the server's operational status and if it cannot determine the status, it then tries to make a connection to the server. If both operations fail, the watchdog stops and then restarts the server.

Before you use the watchdog, verify that it is enabled and running using the `imadmin check` command. By default, the watchdog is enabled and running when you install Instant Messaging.

More information about the `imadmin` utility is available in Appendix B, "Instant Messaging imadmin Tool Reference".

Determining the Status of the Watchdog

You use the `imadmin` command-line utility to check the status of the watchdog.

➤ To Determine the Status of the Watchdog

1. Change to the directory that contains the `imadmin` command-line utility:

   cd *im_svr_base*/sbin

2. Run `imadmin check`:

   ```
   ./imadmin check watchdog
   ```

   The `imadmin` utility returns the current status of the watchdog.

Enabling and Disabling the Watchdog

By default, the watchdog is enabled when you install Instant Messaging. You can disable or enable the watchdog by setting a configuration parameter in `iim.conf`.

➤ To Enable or Disable the Watchdog

1. Change to the directory that contains the `iim.conf` file.

   `cd` *im_cfg_base*

2. Enable or disable the watchdog by setting the `iim_wd.enable` parameter as follows:

   To enable the watchdog: `iim_wd.enable=true`

   To disable the watchdog: `iim_wd.enable=false`

3. Save and close the `iim.conf` file.

4. Refresh the Instant Messaging server configuration:

   `cd` *im_svr_base*`/sbin`

   `./imadmin refresh`

Managing Logging for the Watchdog

You manage logging for the watchdog the same way you manage logging for the server, multiplexor, and the Calendar agent. The watchdog log file is saved as *im_db_base*`/log/iim_wd.log`.

For more information on setting logging levels for all Instant Messaging components including the watchdog, see "Managing Logging" on page 50.

Part III

Reference Information

Appendix A

Instant Messaging Configuration Parameters

This chapter explains the Instant Messaging configuration parameters in the following sections:

Using the iim.conf file

Instant Messaging stores configuration settings in the `iim.conf` file within the Configuration Directory (*im_cfg_base*).

- On Solaris:

  `/etc/opt/SUNWiim/default/config/iim.conf`

- On Linux:

  `/etc/opt/sun/im/default/config/iim.conf`

This file is a plain ASCII text file, with each line defining a server parameter and its value(s):

- A parameter and its value(s) are separated by an equal sign (=) with spaces and tabs allowed before or after the equal sign.
- A value can be enclosed in double quotes (`" "`). If a parameter allows multiple values, the entire value string must be enclosed in double quotes.
- A comment line must have an exclamation point (`!`) as the first character of the line. Comment lines are for informational purposes and are ignored by the server.
- If a parameter appears more than once, the value of the last parameter listed overrides the previous value.
- A backslash (`\`) is used for continuation and indicates the value(s) are longer than one line.
- Each line is terminated by a line terminator (`\n`, `\r`, or `\r\n`).
- The key consists of all the characters in the line starting with the first non-whitespace character and up to the first ASCII equal sign (=) or semi-colon (`;`). If the key is terminated by a semi-colon, it is followed by "`lang-`" and a tag that indicates the language in which this value is to be interpreted. The language tag is followed by an equal sign (=). All whitespace characters before and after the equal sign are ignored. All remaining characters on the line become part of the associated value string.
- Multiple values in the value string are separated using commas (`,`).
- Within a value, if any special characters like comma, space, newline, tab, double quotes, or backslash are present, the entire value needs to be within double quotes. In addition, every carriage return, line feed, tab, backslash, and double quotes within the value must specified with a backslash (\).
- If you make changes to the `iim.conf` file, you must refresh the Instant Messaging server in order for the new configuration settings to take effect.

NOTE The `iim.conf` file is initialized by the installation process and should be modified only as described in this guide.

General Configuration Parameters

Table A-1 lists and describes the general configuration parameters.

Table A-1 General Configuration Parameters

Parameter	Default Value	Description
iim.comm.modules	iim_server,iim_mux	The communication modules used. The possible values are iim_server and iim_mux. The default value is iim_server, iim_mux, which means both the server and multiplexor are used. The iim_mux value is useful for multiplexor.
iim.smtpserver	localhost	SMTP server to send mail to end users who have set the option for forwarding their messages as emails or to pagers.
iim.instancedir	/opt	The installation directory root.
iim.instancevardir	Solaris: /var/opt/SUNWiim Linux: /var/opt/sun/im	Sets the directory to contain runtime files, including the end-user profile database, logs, and other files created by the server and multiplexor at runtime.
iim.user	inetuser for LDAP deployments. root for portal deployment.	The end-user name with which the server processes run.
iim.group	inetgroup for LDAP deployments. root for portal deployment.	The group using which the server processes run.
iim.jvm.maxmemorysize	256	The maximum number heap size in MB the JVM running the server is allowed to use. Used to construct the -mx argument of the Java command.
iim.mail.charset	None	This parameter specifies if the headers of the mail are in ASCII and not encoded. It contains the name of the charset to be used to encode the headers of the mail message sent out for offline alerts. For example: iim.mail.charset=iso-2022-jp
iim.jvm.command	/usr/j2se/bin/java	The location of the Java Runtime Executable (JRE).

Table A-1 General Configuration Parameters *(Continued)*

Parameter	Default Value	Description
`iim.identity.basedir`	`/opt`	The default installation directory, also referred to as the base directory, for Sun Java System Access Manager.
`iim.identity.jre`	`/usr/java_1.3.1_04`	The location of the JRE used by the Access Manager to run all it's processes.
`iim.portal.deployuri`	`/portal`	The URI using which the Portal Server `war` files are deployed in the Access Manager.
`iim.portal.host`	`imhostname`	The host name of the server on which the Portal Server is running. Specify the port number if a non default port number is used.
`iim.portal.protocol`	`http`	The protocol used to access the Portal Server.
`iim.policy.resynctime`	`720`	The Instant Messaging server clears all cached end-user information on a regular basis in order to eliminate old end-user information. This parameter specifies the frequency, in minutes, at which the cached end-user information is cleared.

User Source Configuration Parameters

Table A-2 lists and describes the user source configuration parameters.

Table A-2 User Source Configuration Parameters

Parameter	Default Value	Description
`iim_ldap.host`	`localhost:389`	LDAP server name and port used by Instant Messaging server for end-user authentication.
`iim_ldap.searchbase`	`o=internet`	The string used as base to search for the end users and groups on the LDAP server.
`iim_ldap.usergroupbinddn`	None (the server performs anonymous searches)	Specifies the `dn` to use to bind to the LDAP server for searches.
`iim_ldap.usergroupbindcred`	None (the server performs anonymous searches)	Specifies the password to use with the `iim_ldap.usergroupbinddn` dn for LDAP searches.

Table A-2 User Source Configuration Parameters *(Continued)*

Parameter	Default Value	Description
`iim_ldap.loginfilter`	`(&(|(objectclass=inetorgperson)(objectclass=webtopuser))(uid={0}))`	Search filter used during end-user login.
`iim_ldap.usergroupbyidsearchfilter`	`(|(&(objectclass=groupofuniquenames)(uid={0}))(&(|(objectclass=inetorgperson)(objectclass=webtopuser))(uid={0})))`	The search filter used to search for end users and groups in the directory, under the base specified by ID.

`iim_ldap.usergroupbynamesearchfilter`	`(|(&(objectclass=groupofuniquenames)(cn={0}))`
	`(&(|(objectclass=inetorgperson)(objectclass=webtopuser))(cn={0})))`
		The search filter used to search for end users and groups in the directory, under the base specified by name.

`iim_ldap.allowwildcardinuid`	`False`	Determines if wildcards should be enabled for UIDs while performing a search. As most directory installations have UIDs indexed for exact searches only, the default value is `False`. Setting this value to `True` can impact performance unless UIDs are indexed for substring search.
`iim_ldap.userclass`	`inetOrgPerson,webtopuser`	The LDAP class that indicates that an entry belongs to an end user.
`iim_ldap.groupclass`	`groupOfUniqueNames`	The LDAP class that indicates that an entry belongs to a group.
`iim_ldap.groupbrowsefilter`	`(objectclass=groupofuniquenames)`	The search filter used to browse all groups in the directory, under the specified search base.
`iim_ldap.searchlimit`	`40`	Maximum number of entries to be returned by a search. A value of `-1` means search is disabled on this server and a value of `0` indicates unlimited search.
`iim_ldap.userdisplay`	`cn`	LDAP attribute to use for display name of end users.
`iim_ldap.groupdisplay`	`cn`	LDAP attribute to use for display name of groups.
`im_ldap.useruidattr`	`uid`	LDAP attribute used as end users' UID.
`im_ldap.groupmemberattr`	`uniquemember`	LDAP attribute that gives the list of members of a group.

Table A-2 User Source Configuration Parameters *(Continued)*

Parameter	Default Value	Description
iim_ldap.usermailattr	mail	LDAP attribute that should contain end users' provisioned email addresses. Used when the email message is sent to an offline end user.
iim_ldap.userattributes	None	LDAP attribute that contains the list of custom attributes from the LDAP user entry.
iim_ldap.groupattributes	None	LDAP attribute that contains the list of custom attributes from the LDAP group entry.
iim_ldap.groupmemberurlattr	None	The membership attribute of a dynamic group, which contains the LDAP filter or the LDAP URL.
iim_ldap.useidentityadmin	The default value is true, if Access Manager Instant Messaging Service Definition component is installed. The default value is false, if Access Manager Instant Messaging Service Definition component is not installed.	If the value is true then the Access Manager Administrator credentials will be used to bind to the Directory Server.

Logging Configuration Parameters

Table A-3 lists and describes the logging configuration parameters.

Table A-3 Logging Configuration Parameters

Parameter	Default Value	Description
iim.log.iim_server.severity	INFO	Level of logging required for the server module. The possible values from highest to lowest are: FATAL, ERROR, WARNING, INFO, and DEBUG. If a lower level of logging is chosen, it is implied that you get the higher levels too. That is, if you choose WARNING you get FATAL, ERROR, and WARNING.

Table A-3 Logging Configuration Parameters *(Continued)*

Parameter	Default Value	Description
`iim.log.iim_server.url`	*im_runtime_base*`/log/xmppd.log`	Location of the server log file. This file needs to be periodically trimmed to prevent disk space from filling up.
`iim.log.iim_mux.severity`	`INFO`	Level of logging required for the multiplexor module. The possible values from highest to lowest are: FATAL, ERROR, WARNING, INFO, and DEBUG. If a lower level of logging is chosen, it is implied that you get the higher levels too. That is, if you choose WARNING you get FATAL, ERROR, and WARNING.
`iim.log.iim_mux.url`	*im_runtime_base*`/log/mux.log`	Location of the multiplexor log file. This file needs to be periodically trimmed to prevent disk space from filling up.
`iim.log.iim_server.maxlogsize`		This parameter contains the maximum size of a server log file. If the log files exceeds the size specified in this parameter then server creates a new file to log in the details.
`iim.log.iim_wd.severity`	`INFO`	Level of logging required for the watchdog. The possible values from highest to lowest are: FATAL, ERROR, WARNING, INFO, and DEBUG. If a lower level of logging is chosen, it is implied that you get the higher levels too. That is, if you choose WARNING you get FATAL, ERROR, and WARNING.
`iim.log.iim_calendar.severity`	`INFO`	Level of logging required for the Calendar agent. The possible values from highest to lowest are: FATAL, ERROR, WARNING, INFO, and DEBUG. If a lower level of logging is chosen, it is implied that you get the higher levels too. That is, if you choose WARNING you get FATAL, ERROR, and WARNING.

Instant Messaging Server Configuration Parameters

Table A-4 lists and describes the Instant Messaging server configuration parameters.

Table A-4 General Instant Messaging server Configuration Parameters

Parameter	Default Value	Description
`iim_server.autosubscribe`	`FALSE`	Indicates whether subscriptions are automatically authorized by the server. The possible values are `TRUE` and `FALSE`. If `TRUE`, subscribe requests are automatically followed by a subscribed response generated by the server. The server then sends the modified roster to the subscriber and the user the subscriber added as a contact. The user and the contact must be in the same domain to use this feature.
`iim_server.domainname`	*host's domain name*	The logical Instant Messaging server domain name you want this server to support. This is the name that is used by other servers in the network to identify this server. It is also the name used by this server to identify its end users to other servers. This is not necessarily the Fully Qualified Domain Name of the system running the Instant Messaging server. For example, if the system `iim.xyz.com` is the only Instant Messaging server for a company `xyz.com`, then the domain name is likely to be `xyz.com`.

Table A-4 General Instant Messaging server Configuration Parameters *(Continued)*

Parameter	Default Value	Description
`iim_server.port`	5269	IP address and port for the server to bind to, and to listen for connections from other servers. IP address setting is useful for multi homed machines when you want to use only one particular IP address. If no IP address is listed, this indicates a value of `INADDR_ANY` on `localhost`.
`iim_server.useport`	`TRUE`	Indicates whether the server should listen on the server-to-server communication port. The possible values are `TRUE` and `FALSE`. If `TRUE`, the server listens on the port defined by `iim_server.port` or on port 9919, if that is not explicitly defined.
`iim_server.sslport`	5223	Server's SSL port used for secure server-to-server communication. Note: The value format is `IPaddress:port`. If no IP address is listed, this indicates a value of `INADDR_ANY` on `localhost`.
`iim_server.usesslport`	`FALSE`	Indicates if the server should listen on the server-to-server SSL communication port. The possible values are `TRUE` and `FALSE`. If `TRUE`, the server listens on the port defined by `iim_server.sslport` or on port 5223, if that is not explicitly defined.
`iim_server.clienttimeout`	15	Specifies the time, in minutes, before the server discards client connections that are no longer active. For example, when a machine is turned off. The minimum accepted value is 5.

Table A-4 General Instant Messaging server Configuration Parameters *(Continued)*

Parameter	Default Value	Description
iim_server.usesso	0	This parameter tells the server whether or not depend on the SSO provider during authentication. An SSO provider is a module which the server uses to validate a session id with a SSO service. In portal deployment, Portal Server Session API provides the IM server with the ability to validate session ids sent by the client. The value for this parameter can either be 0, 1,or -1. 0 - do not use the SSO provider (default). 1 - use the SSO provider first and default to LDAP when the SSO validation fails. -1- use SSO provider only without attempting LDAP authentication even when the SSO validation fails. The iim_server.usesso parameter is used in conjunction with the iim_server.ssoprovider parameter.
iim_server.ssoprovider	None	This parameter specifies the class implementing the SSO Provider. If iim_server.usesso is not equal to 0 and this option is not set, the server uses the default Portal Server based SSO Provider.
iim_server.msg_archive	false	This parameter specifies whether the archive provider should be enabled or disabled.

Table A-4 General Instant Messaging server Configuration Parameters *(Continued)*

Parameter	Default Value	Description
`iim_server.msg_archive.provider`	None	This parameter contains the list of custom archive providers. This parameter allows multiple values and each value is separated by a comma(`,`).
`iim_server.msg_archive.auto`	false	This parameter tells the server whether the end-users' archive control settings can be considered. If the value for this parameter is true, it is equivalent to selecting `archive everything` option in the User Settings.
`iim_server.conversion`	`false`	This parameter specifies whether message conversion should be enabled. It specifies whether the configured list of Message Conversion Providers should be used for message conversion.
`iim_server.conversion.provider`	None	This parameter contains the list of Message Conversion Providers to be used for message conversion. This parameter allows multiple values with each value is separated by a comma(`,`).

Table A-4 General Instant Messaging server Configuration Parameters *(Continued)*

Parameter	Default Value	Description
iim_server.servertimeout	-1	The server can be configured to automatically close the connection opened by a remote server, if the remote server is inactive. This is performed by periodically measuring the time the last request was made by the remote server to the server. The connection to the remote server is terminated, if the time of the last request made by the remote server exceeds the value of the iim_server.servertimeout parameter. The parameter value is in minutes.
iim_server.enable	true	This value should contain whether or not the Instant Messaging server should be enabled. This parameter is set false to enable the Instant Messaging multiplexor.
iim_server.conversion.external.command	None	This parameter contains the external command used for message conversion.
iim_server.stat_frequency	1	This parameter contains the frequency at which the server logs the summary of activities to the log file. The server logs the summary of activities to the log file only if the server minimum log severity is set to INFO or lower. The value is in minutes.
iim_server.secconfigdir	/etc/opt/SUNWiim/default/config	This directory contains the key and certificate databases. It usually contains the security module database.

Table A-4 General Instant Messaging server Configuration Parameters *(Continued)*

Parameter	Default Value	Description
`iim_server.keydbprefix`	None	This value should contain the key database filename prefix. The key database file name must always end with `key3.db.` If the Key database contains a prefix, for example `This-Database-key3.db`, then value of this parameter is `This-Database`.
`iim_server.certdbprefix`	None	This value should contain the certificate database filename prefix. The certificate database file name must always end with `cert7.db`. If the certificate database contains a prefix, for example `Secret-stuff-cert7.db`, then value of this parameter is `Secret-stuff`.
`iim_server.secmodfile`	`secmod.db`	This value should contain the name of the security module file.
`iim_server.certnickname`	`Server-Cert`	This value should contain the name of the certificate you entered while installing the certificate. The certificate name is case-sensitive.
`iim_server.keystorepasswordfile`	`sslpassword.conf`	This value should contain the relative path and the name of the file containing the password for the key database. This file should contain the following line: `Internal (Software) Token:`*password* Where *password* is the password protecting the key database.
`iim_server.trust_all_cert`	false	If this value is true than the server will trust all certificates and will also add the certificate information into the log files.

Multiple Server Configuration Parameters

For communication between multiple Instant Messaging servers in your network, you need to configure your server to identify itself with the other servers and identify itself with each coserver, or cooperating server, which will have a connection to your server. The coserver identifies itself with its Instant Messaging domain name, host and port number, serverID, and password.

Each cooperating server is given a symbolic name, which is a string consisting of letters and digits, for example, `coserver1`. Using the symbolic naming convention you can specify multiple servers.

When Instant Messaging servers are configured in this manner, you can form a larger Instant Messaging community. Therefore:

- End users on each server can communicate with end users on every other server
- Use conferences rooms on other servers
- Subscribe to news channels on other servers (subject to access privileges)

Table A-5 lists and describes the multiple server configuration parameters.

Table A-5 Multiple Server Configuration Parameters

Parameter	Default Value	Description
`iim_server.serverid`	None	String used by this server to identify itself to all other servers.
`iim_server.password`	None	Password used by this server to authenticate itself to all other servers.

`iim_server.coservers`	None	Comma separated list containing symbolic names of the servers that can connect to this server. Any meaningful names are allowed, but they must match what you use for the `.serverid`, `.password`, and `.host` parameters. Examples:
		`iim_server.coservers=coserver1,coserver2`
		or
		`iim_server.coservers=abc,xyz,ntc`

Table A-5 Multiple Server Configuration Parameters *(Continued)*

Parameter	Default Value	Description
`iim_server.`*coserver1*`.serverid`	None	String that identifies the cooperating server represented by the name, *coserver1* to authenticate to this server. Note: If you had used `abc` in the `iim_server.coservers` list, then the corresponding name for its *serverid* would be `iim_server.abc.serverid`.
`iim_server.`*coserver1*`.password`	None	Password used by cooperating server represented by the name, *coserver1* to authenticate to this server. Note: If you had used `abc` in the `iim_server.coservers` list, then the corresponding name for its password would be `iim_server.abc.password`.
`iim_server.`*coserver1*`.host`	None	IP address and the port to connect to, for end users on this server to communicate to end users on the server represented by the name `coserver1`. Note: If you had used `abc` in the `iim_server.coservers` list, then the corresponding name for its host would be `iim_server.abc.host`. Note: The value format is `name:port` or `IPaddress:port`.
`iim_server.`*coserver1*`.usessl`	`False`	Indicates if this server should use SSL to talk to the server identified by *coserver1*. The possible values are `TRUE` and `FALSE`.

Multiplexor Configuration Parameters

Table A-6 lists and describes the multiplexor configuration parameters.

Table A-6 Multiplexor Configuration Parameters

Parameter	Default Value	Description
`iim_mux.listenport`	*multiplexorname or IP address:*5222	IP address or FQDN and listening port on which the multiplexor listens for incoming requests from Instant Messenger. The value format is *IP_address:port* or *multiplexorname:port*. If no IP address or domain name is listed, this indicates a value of `INADDR_ANY` on localhost. If you change this value, also change the `im.html` and `im.jnlp` files so that they match the port value.
`iim_mux.serverport`	45222	The IM server and port the multiplexor talks to. The value format is *servername*`:`*port* or *IP_address*`:`*port*.
`iim_mux.numinstances`	1	Number of instances of the multiplexor. This parameter is valid only for Solaris platforms.
`iim_mux.maxthreads`	5	Maximum number of threads per instance of the multiplexor.
`iim_mux.maxsessions`	2000	Maximum number of concurrent connections per multiplexor process.
`iim_mux.usessl`	off	If the value is set to `on`, the multiplexor requires an SSL handshake for each connection it accepts, before exchanging any application data.
`iim_mux.secconfigdir`	`/etc/opt/SUNWiim/default/config`	The `/etc/opt/SUNWiim/default/config` is the value of the `iim_mux.secconfigdir` parameter. This directory contains the key and certificate databases. It usually contains the security module database.
`iim_mux.keydbprefix`	None	This value should contain the key database filename prefix. The key database file name must always end with `key3.db.` If the Key database contains a prefix, for example `This-Database-key3.db`, then value of this parameter is `This-Database`.

Table A-6 Multiplexor Configuration Parameters *(Continued)*

Parameter	Default Value	Description
`iim_mux.certdbprefix`	None	This value should contain the certificate database filename prefix. The certificate database file name must always end with `cert7.db`. If the certificate database contains a prefix, for example `Secret-stuff-cert7.db`, then value of this parameter is `Secret-stuff`.
`iim_mux.secmodfile`	`secmod.db`	This value should contain the name of the security module file.
`iim_mux.certnickname`	`Server-Cert`	This value should contain the name of the certificate you entered while installing the certificate. The certificate name is case-sensitive.
`iim_mux.keystorepasswordfile`	`/etc/opt/SUNWiim/default/config/sslpassword.conf`	This value should contain the relative path and the name of the file containing the password for the key database. This file should contain the following line: `Internal (Software) Token:`*password* Where *password* is the password protecting the key database.
`iim_mux.stat_frequency`	`600`	This value should contain the frequency at which the multiplexor logs the summary of activities to the log file.The minimum value is 10 seconds.
`iim_mux.enable`	true	If the value is true then the multiplexor will run for this instance. If the value is false then the multiplexor will not run for this instance.

Watchdog Parameters

The watchdog monitors the server process and attempts to restart the server if it determines that the server is not running. See "Managing the Watchdog Process" on page 157 for more information about the watchdog.

Table A-7 lists and describes the watchdog configuration parameters.

Table A-7 Watchdog Configuration Parameters

Parameter	Default Value	Description
`iim_wd.enable`	`true`	Enables the watchdog feature. To reset this parameter or disable the watchdog, set this to false. To avoid conflicts, you should disable the watchdog if you are monitoring the Instant Messaging server using the operating system administration console.
`iim_wd.period`	`300` (seconds)	The watchdog periodically polls the server to check whether it is running. This parameter sets the interval between two status polls.
`iim_wd.maxRetries`	3 (retries)	Sets the number of times the watchdog will attempt to contact the Instant Messaging server before shutting down and restarting the server. The maximum is ten retries.

Agent Parameters

Agents, such as the Calendar agent, enable functionality within the Instant Messaging server and enhance its interoperability with other Sun Java™ System servers.

Table A-8 lists and describes agent configuration parameters.

Table A-8 Agent Configuration Parameters

Parameter	Default Value	Description
`jms.consumers`	None	Used with the Calendar agent. Contains the name of the alarm. The value for this parameter must be set to: `cal_reminder`

Table A-8 Agent Configuration Parameters *(Continued)*

Parameter	Default Value	Description
`jms.consumer.cal_reminder.destination`	None	Used with the Calendar agent. Destination of the alarm. This must be the same as the value of the `caldb.serveralarms.url` configuration parameter in the `ics.conf` file. For example, `enp:///ics/customalarm`
`jms.consumer.cal_reminder.provider`	None	Used with the Calendar agent. The name of the provider. Typically, this is set to `ens`. The value for this parameter must be the same as the name in `jms.providers`.
`jms.consumer.cal_reminder.type`	None	Used with the Calendar agent. The type of alarm to set. The value for this parameter must be set to: `topic`
`jms.consumer.cal_reminder.param`	None	Used with the Calendar agent. The alarm parameter. The value for this parameter must be set as follows including the quotes: `"eventtype=calendar.alarm"`
`jms.consumer.cal_reminder.factory`	None	Used with the Calendar agent. A listener that registers itself for the new calendar reminder messages. The value for this parameter must be set to: `com.iplanet.im.server.JMSCalendarMessageListener`
`jms.providers`	None	Used with the Calendar agent. The name of the provider. Typically, you set the value of this parameter to `ens`. This must be the same as the value listed for jms.consumer.cal_reminder.provider.
`jms.provider.ens.broker`	None	Used with the Calendar agent. Hostname of the ENS and the port number on which the ENS listens for incoming requests. Set to the port specified in the `ics.conf` file parameter `service.ens.port`. The default is 57997. For example: `jms.provider.ens.broker=cal.example.com:57997`
`jms.provider.ens.factory`	None	Used with the Calendar agent. Factory class used for creating the topic connection objects. The value for this parameter must be set to: `com.iplanet.ens.jms.EnsTopicConnFactory`

Table A-8 Agent Configuration Parameters *(Continued)*

Parameter	Default Value	Description
iim_agent.enable	None	If TRUE or absent from iim.conf, enables Instant Messaging agents. Set the value to FALSE to disable all agents.
iim_agent.agent-calendar.enable	None	Used with the Calendar agent. If TRUE or absent from iim.conf, loads a component that enables the Calendar agent specifically.
agent-calendar.jid	None	The JID of the Calendar agent.
agent-calendar.password	None	The Calendar agent password.
iim_server.components	None	Describes the Calendar agent as a component of the Instant Messaging server. The value of this parameter must be set to: agent-calendar

Appendix B

Instant Messaging imadmin Tool Reference

This chapter explains the `imadmin` command used to administer Instant Messaging.

imadmin

You can use the `imadmin` utility to start, stop, and refresh the Instant Messaging server and multiplexor. Run `imadmin` as `root` or as the end user you specified during configuration.

Requirements:

You must invoke the `imadmin` utility from the host on which Instant Messaging server is installed.

Location:

im_svr_base/`sbin`

Table B-1 lists and describes commands related to the `imadmin` command.

Table B-1 imadmin Commands with Descriptions

Command	Description
`imadmin check`	Checks to see if the components (`server`, `multiplexor`, `agent-calendar`, and `watchdog`) are up and running and displays the results. If you don't specify a component, the `imadmin` utility returns information about all components.
`imadmin start`	Starts the enabled server and/or multiplexor component(s).

Table B-1 imadmin Commands with Descriptions *(Continued)*

Command	Description
`imadmin stop`	Stops the enabled server and/or multiplexor component(s).
`imadmin refresh`	Refreshes the enabled server and/or multiplexor component(s).
`imadmin start server`	Starts only the server.
`imadmin stop server`	Stops only the server.
`imadmin refresh server`	Refreshes only the server.
`imadmin start multiplexor`	Starts only the multiplexor.
`imadmin stop multiplexor`	Stops only the multiplexor.
`imadmin refresh multiplexor`	Refreshes only the multiplexor.
`imadmin start agent-calendar`	Starts only the Calendar agent.
`imadmin stop agent-calendar`	Stops only the Calendar agent.
`imadmin refresh agent-calendar`	Refreshes only the Calendar agent.
`imadmin start watchdog`	Starts only the watchdog.
`imadmin stop watchdog`	Stops only the watchdog.
`imadmin refresh watchdog`	Refreshes only the watchdog.
`imadmin migrate`	Generates Sun Java System Access Manager policies based on current policy access control files. See the *Sun Java Enterprise System 2005Q1Upgrade and Migration Guide* at: `http://docs.sun.com/doc/819-0062`
`imadmin version`	Displays the version

Synopsis

`imadmin` [*Solaris_options*] [*action*] [*component*]

Options

Table B-2 lists and describes options for the `imadmin` command.

Table B-2 Options for imadmin command

Option	Description
`-c` *alt-config-file*	Used with the `start` and `refresh` actions, to specify a different configuration file other than `/etc/opt/SUNWiim/config/iim.conf` file
`-h`	Displays help on the `imadmin` command.

Actions

Table B-3 lists and describes actions performed after various `imadmin` commands are issued.

Table B-3 Actions for imadmin Command

Option	Description
check	Returns information about Instant Messaging components (`server`, `multiplexor`, `agent-calendar`, and `watchdog`). You do not need to provide a *[component]* with this action.
`start`	Sets the `classpath`, the Java heap size and starts all the specified components.
`stop`	Stops all the specified component's daemons.
`refresh`	Stops and starts the specified component(s). Useful after a configuration change.

Components

Table B-4 lists and describes the components for the `imadmin` command.

Table B-4 Components for imadmin Command

Option	Description
`agent-calendar`	Indicates the Calendar agent (`agent-calendar`).
`multiplexor`	Indicates the multiplexor alone.
`server`	Indicates the Instant Messaging server.
`watchdog`	Indicates the watchdog.

Appendix C

Instant Messaging APIs

This chapter describes the APIs used by Instant Messaging.

Instant Messaging APIs Overview

Instant Messaging provides Java APIs which can be used to develop extension or integration modules. Detailed documentation of these APIs are provided with the installed Instant Messenger component, in the form of HTML files generated by Javadocs. The Javadoc files are installed in the *im_svr_base*`/html/apidocs/` directory. To view the API documentation, point your browser to *imcodebase*`/apidocs` where the `codebase` is the Instant Messenger resources codebase.

The following are the Instant Messaging APIs:

- Instant Messaging Service API
- Messenger Beans
- Service Provider Interfaces
- Authentication Provider API

Instant Messaging Service API

The Instant Messaging API is used by the applications located on the same host or in the remote host to access Instant Messaging services, such as Presence, Conference, Notification, Polls and News channels.

The Instant Messaging Service API can be used for:

- A Java-based or web-based client, such as a portal channel
- A Bridge or a Gateway to enable another class of clients.
- Integration of Instant Messenger and Presence in to the existing applications
- Displaying news feeds as Instant Messenger news.

Messenger Beans

A Messenger bean is a dynamically loaded module used to extend the messenger functionality. Messenger beans can add action listeners, such as buttons and menu items, and item listeners, such as check boxes and toggle buttons in the existing Instant Messenger window. The item listeners are invoked when an end-user input is received and bean-specific actions are based on the end-user input. Beans have the ability to add their own settings panel and save bean-specific properties on the server. Beans can be notified of any event received by the Instant Messenger. for example, a new alert message.

The applications that use Messenger Beans are:

- Ability for end users to share application and conference along with voice or video.
- Ability to retrieve and process the transcript of a conference For example, the contents of a received or sent alert, for archiving purposes.

NOTE The Instant Messenger Archive control functionality is provided through a Messenger Bean.

Service Provider Interfaces

The Service Provider Interface APIs provide the ability to extend the Instant Messaging server functionality. The Service Provider Interface is composed of the following independent APIs:

- The Archive Provider API
- The Document Converter API
- The Authentication Provider API

Archive Provider API

An Archive Provider is a software module usually providing integration with the archive or auditing system. Each configured Archive Provider is invoked for each server process.

The Archive Provider is invoked for the following server processes:

- When a instant message is sent. The Instant Messages, such as alert, poll, chat, news or conference.
- During an authentication event, such as login or logout.
- When there is a change in the presence status.
- During a subscription event. For example, when someone joins or leaves a conference, or subscribes or unsubscribes to a news channel.

The applications that use the Archive Provider API are as follows:

- Instant Messaging Archive

  The default Instant Messaging archive in Instant Messaging is based on the Archive Provider API. For more information on Instant Messaging Archive, see Managing The Instant Messaging Archive.

- The application that records the usage statistics for sizing purposes

Message Conversion API

A Message Converter is invoked for every message or each message part going through the server. The Message Converter may leave the message part intact or modify or remove the message part. The text parts are processed as Java String Objects. The Message Converter processes other attachment as a stream of bytes and returns a potentially different stream of bytes, or nothing at all if the attachment is to be removed.

The applications that uses Message Conversion API are:

- Virus checking and removal
- Translation engine integration
- Message content filtering

Authentication Provider API

The Authentication Provider API provides ability to deploy Instant Messaging in environments that are not using Access Manager password-based or token-based authentication service. This API is invoked whenever an end user requests authentication, and it can be used in conjunction with the LDAP authentication.

Single Sign-on (SSO) with Access Manager is performed using the Authentication Provider API. This API can also be used to integrate with other authentication systems.

Glossary

Refer to the *Java Enterprise System Glossary* (`http://docs.sun.com/doc/816-6873`) for a complete list of terms that are used in this documentation set.

Index