Administrator’s Guide

iPlanet™ Application Server

Version 6.0

816-1677-01
June 2001

THIS 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.

CETTE DOCUMENTATION EST FOURNIE “EN L'ÉTAT”, ET TOUTES CONDITIONS EXPRESSES OU IMPLICITES, TOUTES REPRÉSENTATIONS ET TOUTES GARANTIES, Y COMPRIS TOUTE GARANTIE IMPLICITE D'APTITUDE À LA VENTE, OU À UN BUT PARTICULIER OU DE NON CONTREFAÇON SONT EXCLUES, EXCEPTÉ DANS LA MESURE OÙ DE TELLES EXCLUSIONS SERAIENT CONTRAIRES À LA LOI.

Contents

List of Figures

List of Tables

Preface

This preface contains the following topics:

- Preface
- About This Guide
- How This Guide Is Organized
- Documentation Conventions

Using the Documentation

The following table lists the tasks and concepts that are described in the iPlanet Application Server manuals and *Release Notes*. If you are trying to accomplish a specific task or learn more about a specific concept, refer to the appropriate manual.

Note that the printed manuals are also available online in PDF and HTML format, at: `http://docs.iplanet.com/docs/manuals/ias.html`.

For information about	**See the following**	**Shipped with**
Late-breaking information about the software and the documentation	Release Notes	iPlanet Application Server 6.0
Installing iPlanet Application Server and its various components (Web Connector plug-in, iPlanet Application Server Administrator), and configuring the sample applications	*Installation Guide*	iPlanet Application Server 6.0

For information about	See the following	Shipped with
Creating iPlanet Application Server 6.0 applications that follow the open Java standards model (Servlets, EJBs, JSPs, and JDBC), by performing the following tasks: • Creating the presentation and execution layers of an application • Placing discrete pieces of business logic and entities into Enterprise Java Bean (EJB) components • Using JDBC to communicate with databases • Using iterative testing, debugging, and application fine-tuning procedures to generate applications that execute correctly and quickly	*Developer's Guide (Java)*	iPlanet Application Server 6.0

For information about	See the following	Shipped with
Administering one or more application servers using the iPlanet Application Server Administrator Tool to perform the following tasks: • Monitoring and logging server activity • Implementing security for iPlanet Application Server • Enabling high availability of server resources • Configuring web-connector plugin • Administering database connectivity • Administering transactions • Configuring multiple servers • Administering multiple-server applications • Load balancing servers • Managing distributed data synchronization • Setting up iPlanet Application Server for development	*Administrator's Guide*	iPlanet Application Server 6.0
Migrating your applications to the new iPlanet Application Server 6.0 programming model from the Netscape Application Server version 2.1, including a sample migration of an Online Bank application provided with iPlanet Application Server	*Migration Guide*	iPlanet Application Server 6.0
Using the public classes and interfaces, and their methods in the iPlanet Application Server class library to write C++ applications	*Server Foundation Class Reference (C++)*	Order separately

About This Guide

The *Administrator's Guide* guide leads you through the tasks that you perform as the administrator of one or more iPlanet Application Server machines. This guide assumes that you have installed iPlanet Application Server on at least one machine. For information about installing iPlanet Application Server, refer to the *Installation Guide*.

You perform most of the administration tasks with iPlanet Application Server Administration Tool, a GUI-based tool for server and application administration. This tool is described in About iPlanet Application Server Administration Tool (iASAT).

How This Guide Is Organized

This guide is divided into three parts. If you are new to administering an iPlanet Application Server machine, begin with Part I, Getting Started for an overview of how to start the server and Administration Tool. If you are already familiar with administering application servers, skim Part I, before going on to Part II, Administering a Single iPlanet Application Server.

If you are administering more than one application server, continue to Part III, Administering Multiple iPlanet Application Servers, for additional, multiple-server enterprise specific information.

Part I: Getting Started

The first part of the *Administrator's Guide* describes the environment of iPlanet Application Server.

The following chapter is included in this part:

- Chapter 1, Performing Basic Administrative Tasks, describes how to get started with iPlanet Application Server Administration Tool, as well as the basic iPlanet Application Server configuration tasks that you can perform.

Part II: Administering a Single iPlanet Application Server

The second part of the *Administrator's Guide* describes server and application administration procedures for a single iPlanet Application Server machine.

The following chapters are included in this part:

- Chapter 2, Monitoring Server Activity, describes the monitoring service that allows you to chart various attributes of the Executive, Java, and C++ server processes.
- Chapter 3, Configuring SNMP to Monitor iPlanet Application Server with Third-Party Tools, describes how to configure Simple Network Management Protocol (SNMP) so you can monitor iPlanet Application Server with a third-party SNMP management tool.
- Chapter 4, Logging Server Messages, describes the message-logging service provided by iPlanet Application Server.
- Chapter 5, Securing iPlanet Application Server, describes how to set up users and groups to provide security for your applications.
- Chapter 6, Enabling High Availability of Server Resources, describes how you can increase application performance.
- Chapter 7, Configuring the Web Connector Plug-In, describes the web connector plug-in, which sends users' requests to applications residing on iPlanet Application Server.
- Chapter 8, Administering Database Connectivity, describes how to configure data access drivers and apply settings to database connectivity parameters.
- Chapter 9, Administering Transactions, describes the tasks and conceptual information necessary for administering transactions using iPlanet Application Server Administration Tool.

Part III: Administering Multiple iPlanet Application Servers

The third part of the *Administrator's Guide* describes how to administer multiple iPlanet Application Server machines. More in-depth administration procedures and concepts that apply to a multiple-server enterprise have been included. These procedures focus solely on multiple-server administration, and are used along with the single-server procedures described in Part II.

The following chapters are included in this part:

- Chapter 10, Configuring Multiple Servers, describes how to configure the web connector plug-in, distributed data synchronization, and multicast communication for multiple iPlanet Application Server machines, using iPlanet Application Server Administration Tool.
- Chapter 11, Administering Multi-Server Applications, describes how to maintain multiple iPlanet Application Server machines at the same time using iPlanet Application Server Administration Tool.
- Chapter 12, Balancing User-Request Loads, describes load balancing, which optimizes the ability of each iPlanet Application Server machine to process users' requests by keeping those requests balanced among several application servers.
- Chapter 13, Managing Distributed Data Synchronization, describes how to group iPlanet Application Server machines into data synchronization clusters.
- Chapter 14, Setting Up iPlanet Application Server for Development, describes how to set up iPlanet Application Server for the purpose of developing applications. This section would be of special interest to developers.
- Appendix A, Troubleshooting contains troubleshooting information about your iPlanet Application Server machine.

Documentation Conventions

This guide uses the following documentation conventions:

Conventions	Description

File and directory paths

Windows: Directory path names are separated using Backslashes. For example, Images\Screens\ser.gif.

Solaris: The directory paths are separated using forward slashes. For example, Images/Screens/ser.gif.

Menu options

Second level menu options are separated by the ‘greater than’ symbol (>). For example, invoking an application from the Start menu in Windows is described this way:

From the Start menu, choose Programs>iPlanet Application Server 6.0. Then select iPlanet Administration Tool.

URLs

This guide uses URLs of the form:

http://*server.domain:port*/*path*/*file*.html

Where,

- *server* is the name of server on which you run your application
- *domain* is your Internet domain name
- *path* is the directory structure on the server
- *file* is an individual filename.

Note: Italicized items in URLs are placeholders.

Fonts

`The monospace` font is used for:

- sample code and code listings
- API and language elements (such as function names and class names)
- file names, path names, directory names, and HTML tags.

Italic type is used for:

- book titles
- variables and placeholders

Part 1

Getting Started

Chapter 1

Performing Basic Administrative Tasks

This chapter describes how to administer iPlanet Application Server, using iPlanet Application Server Administration Tool (iASAT), iPlanet Registry Editor and command line tools.

This chapter also covers the basic iPlanet Application Server configuration tasks you can perform using iPlanet Application Server Administration Tool (iASAT), various command line tools and iPlanet Registry Editor.

The basic administrative and configuration tasks that you can perform using iASAT, command-line tools and iPlanet Registry Editor are divided into the following topics:

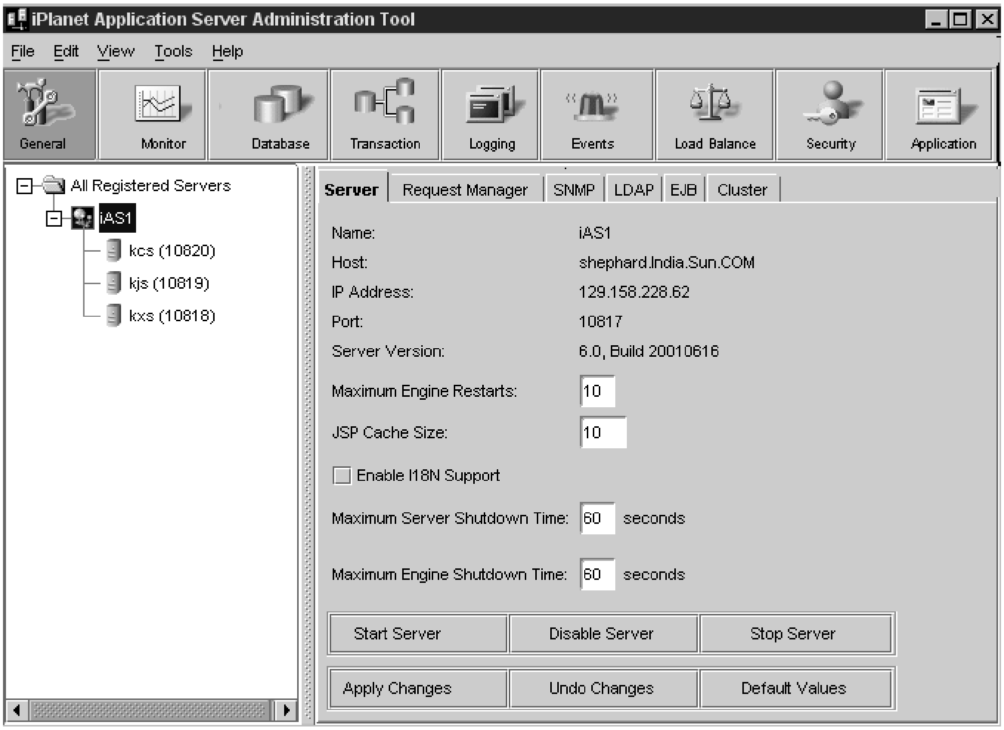
- About iPlanet Application Server Administration Tool (iASAT)
- Performing Administrative Tasks Using iASAT
- About Command-Line Tools
- Using Command-Line Tools
- Performing Administrative Tasks Using Command-Line Tools
- About iPlanet Registry Editor
- Performing Administrative Tasks Using iPlanet Registry Editor

About iPlanet Application Server Administration Tool (iASAT)

iPlanet Application Server Administration Tool (iASAT) is a stand-alone Java application with a graphical user interface that allows you to administer one or more instances of iPlanet Application Server. iPlanet Application Server administration involves performance-related tasks such as adjusting database connection threads and load-balancing parameters. Server administrators must also separately configure components that the application server uses, including the web server.

You also use iASAT to administer application components. You can group, enable and partition Application components using iASAT, to achieve better performance. Application components, the core of an iPlanet Application Server application, are contained in code and stored on the application server. Enterprise Java Beans (EJBs), servlets, JavaServer Pages (JSPs), and AppLogic objects are all application components. For more information about each of these, refer to the *Developer's Guide (Java)*.

Administrative tasks are performed using iASAT. When iASAT is opened to the default General window, the toolbar, main window with left and right panels, and the menu bar are shown as illustrated in the following figure:



Name: iAS1

Host: shephard.India.Sun.COM

IP Address: 129.158.228.62

Port: 10817

Server Version: 6.0, Build 20010616

Maximum Engine Restarts: 10

JSP Cache Size: 10

□ Enable I18N Support

Maximum Server Shutdown Time: 60 seconds

Maximum Engine Shutdown Time: 60 seconds

The left panel of the main window displays all iPlanet Application Servers registered with iASAT. The right panel displays individual features of the registered iPlanet Application Servers.

To Start iASAT

To administer one or more iPlanet Application Server machines, start iASAT using one of the following ways:

- On Windows: From the Start menu, choose `Programs> iPlanet Application Server 6.0> iPlanet Application Server Administration Tool`.
- On a Solaris system: Open a Terminal window and navigate to the path `<iASInstallDir>/ias/bin/` and type the following at the command prompt:

```
ksvradmin &
```

Performing Administrative Tasks Using iASAT

This section describes the concepts and tasks associated with using iASAT. The following topics are included in this section:

- To Register an iPlanet Application Server
- To Unregister an iPlanet Application Server
- To Start and Stop a Server Using iASAT

To Register an iPlanet Application Server

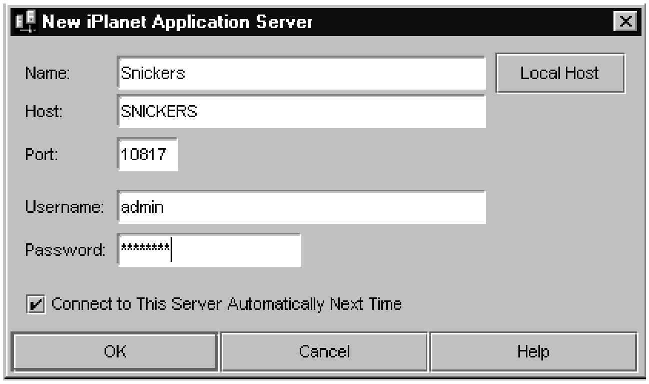
Registering an iPlanet Application Server adds that server to the scope of the Administration Tool. This is best done after you add a server or a group of servers to the enterprise.

NOTE iPlanet Application Server must be registered with iASAT, before you can administer it.

To register an iPlanet Application Server machine, perform the following tasks:

1. On the iASAT toolbar, click General to open the General window.
2. From the File menu, click `New>Server`.

   The New iPlanet Application Server dialog box appears.



New iPlanet Application Server

Name: Snickers

Host: SNICKERS

Port: 10817

Username: admin

Password: ********

☑ Connect to This Server Automatically Next Time

3. Complete one of the following:
   - In the Name text box, specify the name of the server.

This is an arbitrary name you use to distinguish one server from another. For instance, you might name the servers in your enterprise according to their host name, such as Bozo1, Bozo2, Bozo3.

- Click Local Host to register a server running on your local machine.

  This automatically enters a server name and your machine name.

4. In the Host text box, specify the host name of the server.

   This is the DNS name of your server machine. You can also use an IP address.

5. In the Port text box, specify the port number for the Administrative Server. During installation this is set to 10817 by default.

6. If you are registering your local host, in the User Name and Password text boxes, specify the user name and password you entered during installation of the server or when modifying the Users and Groups.

   If the server that you are registering is not the local host, then you need to provide the user name and login password of the machine you want to add. For example, to register a machine called Solo that is available on your network, you need to provide the user name and password of Solo in the User Name and Password text boxes.

7. (Optional) To always connect to the server and display it in the Enterprise window, mark the “Connect to this server” checkbox.

8. Click OK to register the server.

To Unregister an iPlanet Application Server

When a server is no longer available, you can remove it from the scope of the enterprise.

To unregister, or delete an iPlanet Application Server machine, perform the following tasks:

1. On the iASAT toolbar, click General to open the General window.

2. In the left pane of the General window, double-click All Registered Servers.

   A list of all registered servers in the enterprise appears.

3. Select the server you want to delete.

4. From the Edit menu, choose Delete.

   The selected server is removed from the scope of iASAT.

To Start and Stop a Server Using iASAT

You can choose automatic server start-up when you install iPlanet Application Server. Thereafter, iPlanet Application Server starts automatically on system start-up. However, if you manually stop iPlanet Application Server or if the server crashes, you can start the server from iASAT by performing the following tasks:

1. Click General on the iASAT toolbar to open the General window.
2. In the left pane of the General window, select the server you want to start.
3. In the right pane of the General window, click Start Server.

To stop a server, select the server you want to stop and click Stop Server from the right pane of the General window.

NOTE You can expand the servers in the hierarchical tree only when they are running

About Command-Line Tools

iPlanet Application Server comes with various command-line tools and executables, that can be run from the command-line prompt (Windows) and the Shell prompt (Solaris).

Using command-line tools, you can perform a variety of tasks, right from basic configuration to deploying an application.

For a complete list of all the command-line tools that you can use to administer iPlanet Application Server, see Administering Distributed Transactions from the Command Line.

To get a complete description of any command-line tool, type the command at the prompt, insert a space and type `-help`. For example, to get a complete list of all the options that you can try with the iascontrol command, type `iascontrol-help` at the command line prompt.

Using Command-Line Tools

The usage of iPlanet Application Server's command-line tools is different for Windows and Solaris platforms. Most of the command-line tools have been integrated with the iPlanet Application Server Administration Tool and the iPlanet Application Server Deployment Tool, for ease of use.

On Solaris, even though the iPlanet Application Server Administration Tool and the iPlanet Application Server Deployment Tools are available, the usage of command-line tools is quite extensive. As you can have multiple instances of iPlanet Application Server on Solaris, it becomes necessary at times to execute command-line tools from the installation directory of a specific instance of iPlanet Application Server, to modify the attributes of only that instance.

On Windows, command-line tools are in the form of executable (`.exe`) files.

Command-line tools are located in the `<iASInstallDir>/ias/bin` path, on both Solaris and Windows systems.

There are many command-line tools that can be used to administer iPlanet Application Server. These are described in different chapters in this guide. The following section describes how you can stop and start iPlanet Application Server using command-line tools.

Performing Administrative Tasks Using Command-Line Tools

In this section, the procedure for stopping and starting iPlanet Application Server using command-line tools is discussed.

You can stop or start a server using command line tools, on both Windows and Solaris systems, by performing the following tasks:

- On Windows, choose Run from the Start menu.
- On a Solaris systems, navigate to the `<iASInstallDir>\ias\bin\` path.

Run the following script:

```
iascontrol subcommand -instance<instance> -user<user> -password
<password> -host<host> -port<port>
```

where *subcommand* can be one of the following:

`start` Starts an application server instance. If you execute this command from the local host, it also starts the Administration Server (KAS), if not already started.

`stop` Stops the engines of an application server. The administration server is not stopped.

`kill` Forces immediate, non-graceful termination of all application server processes that are running on the local host, as well as the iPlanet Application Server Administration Server (KAS).

The following parameters apply:

`instance` Name of the server instance that you want to stop or start. The server instance must be registered in iASAT.

`user` Name of the user that is authorized to start, stop or kill the server instance. The user must be registered with iASAT

`password` The password associated with the user

`host` The hostname or IP address of the server instance you want to start, stop or kill

`port` The port number of the application server's administrative server. Port 10817 is the default

For example, the command to stop an iPlanet Application Server instance should look like this:

```
iascontrol stop -instance ias2 -user ar126587 -password suniplanet
-host bozo -port 10817
```

NOTE

- You need to register the iPlanet Application Server instance using the Administration Tool, before you use any of these commands.
- If the User ID is longer than 8 characters, the `-stop` and `-kill` commands do not work as expected.

About iPlanet Registry Editor

The iPlanet Registry Editor is a stand-alone GUI tool that displays registry information for iPlanet products. The editor is installed with each instance of iPlanet Application Server and is similar in appearance and function to the registry editor installed on Windows. iPlanet Registry Editor displays values that are stored both in your local machine's registry and your Directory Server.

To Start iPlanet Registry Editor

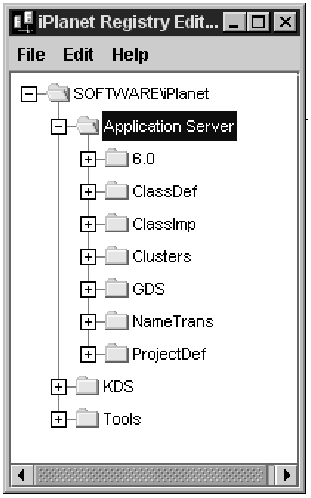
To start iPlanet Registry Editor (known as `kregedit`), perform the following tasks:

1. On Solaris machines, go to `<iASInstallDir>/ias/bin` and type `kregedit` at the command line to launch iPlanet Registry Editor.

NOTE As Solaris supports multiple instances of iPlanet Application Server, each instance has its own registry. Ensure that you navigate to the path where the instance you want is installed, before invoking iPlanet Registry.

1. On Windows systems, click `Start>Run`. Type `kregedit` and click OK.

   The following window appears:



Performing Administrative Tasks Using iPlanet Registry Editor

Using iPlanet Registry, you can effectively manage the entries that form the backend resource for the various functions and processes of iPlanet Application Server. You can add keys, modify key values, and delete redundant keys. iPlanet Registry Editor comes with a comprehensive Find/Replace option using which you find and replace multiple occurrences of the same value in iPlanet Registry.

The iPlanet Application Server Administration Tool and Deployment Tool create back-end entries in iPlanet Registry when required. However, you may at times need to manually create or edit specific configurations in iPlanet Registry, other than those done by iPlanet Application Server tools.

Many of these configuration tasks are described in several chapters in this guide. In this chapter, the following basic configuration tasks that you can perform using iPlanet Registry Editor are discussed:

The following sections describe how to perform basic administrative tasks using iPlanet Registry Editor

- To Add a Value To iPlanet Registry
- To Modify a Value in iPlanet Registry
- To Find and Replace Values In iPlanet Registry
- To Update the Installation Key
- Changing the IP Address
- To Configure Support for Multiple LDAP Domains

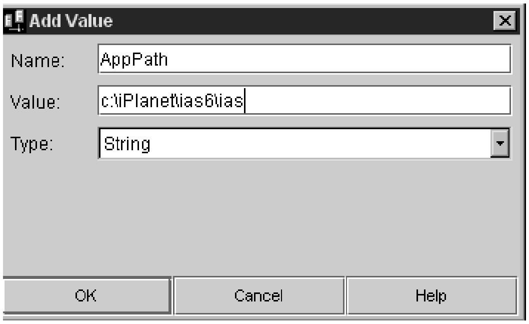
To Add a Value To iPlanet Registry

iPlanet Registry contains several keys, each representing a folder. The values that are contained in these keys are accessed by various tools and servers such as the Administration Tool, Deployment Tool, Web-connector Plug-in and Directory Server.

To add a key to iPlanet Registry, perform the following tasks:

1. Select the key under which you want to create a new key.
2. Click `Edit>Add Key`. Provide a name for the key and click OK.

3. To add a value to this key, select the key and from the Edit menu, choose `Add Value`. The Add Value dialog box appears as shown below:



Add Value

Name: AppPath

Value: c:\iPlanet\ias6\ias

Type: String

4. In the Name field, specify a name for the value.

   Example 1: If you are creating a value that will point to an application path, type AppPath.

   Example 2: If you are specifying the number of auto starts for a server, type AutoStart.

5. In the Value field, provide a value for the key, as required.

   If you are specifying a class path, as indicated in example 1 above, type in the class path, that looks similar to the following example:

   `c:\iplanet\ias6\ias`.

   If you are specifying the number of auto starts for a server, as given in example 2 above, type a number, for example, `1`.

6. In the Type field, select either String or Integer, based on the type of the value you have specified.

   For example, if the value is a class path, select String from the Type drop-down list.

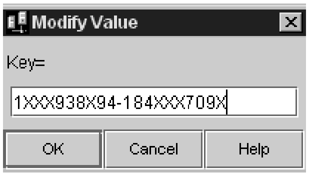
   If the value is a number that indicates, for example, the number of auto restarts specified for a server, select Integer from the Type drop-down list.

7. Click OK to register the value in iPlanet Registry.

To Modify a Value in iPlanet Registry

To modify a value in iPlanet Registry, perform the following tasks:

1. Select the value that you want to modify.
2. Double-click the entry or click `Edit>Modify Value.` The following dialog box appears:



Key= 1XXX938X94-184XXX709X

3. Modify the value and click OK.

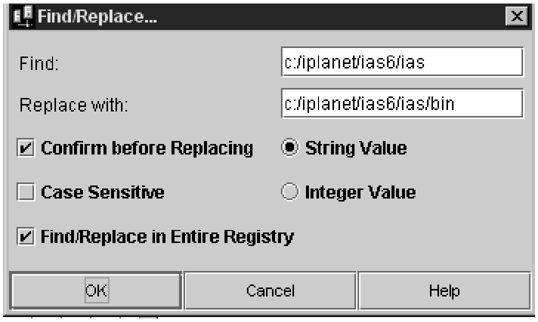
NOTE In the Modify Value dialog box, you can modify only the value specified for the key. You can't modify the type of value, which could be either String or Integer. If you want to modify the value type specification, you need to delete the value and add it again.

To delete a value from the registry, select the value and press Delete or click `Edit>Delete`. You will see a dialog box asking for delete confirmation. Click Yes to confirm the action.

NOTE Once a value is deleted, it cannot be recovered.

To Find and Replace Values In iPlanet Registry

You can find and replace values in iPlanet Registry, using the Find / Replace menu option. You can find and replace a value that occurs within a particular sub-tree or replace all the occurrences of that value in the entire registry.



Find: c:/iplanet/ias6/ias

Replace with: c:/iplanet/ias6/ias/bin

☑ Confirm before Replacing ● String Value

□ Case Sensitive ○ Integer Value

☑ Find/Replace in Entire Registry

To find and replace a value in iPlanet Registry, perform the following tasks:

1. From the Edit menu, choose `Find/Replace`.
2. In the Find text field, specify the value that you want to find in Iplanet Registry. If you want to change only a part of the value, specify only that part. For example, if you want to replace `iPlanet.com` with `Sun.com`, you can type just `iPlanet`.
3. In the Replace with field, specify the value with which you want to replace the existing value. For example, to replace `iPlanet. com` with `Sun.com`, type just Sun. The rest of the value will automatically be retained.
4. Click String, if the value you are searching for is a string, such as a classpath.
5. Click Integer, if the value you are searching for is an integer, such as 1.
6. Mark the Case Sensitive text box, to search for occurrence(s) of the required item that conform to the format of your find input.
7. Mark the Confirm before Replacing checkbox, if you want to be prompted for confirmation before the required value is replaced in iPlanet Registry. When you choose this option, you will be asked for confirmation before the any value is replaced in iPlanet Registry.

NOTE iPlanet Registry holds a lot of critical information on which the various engines and processes depend. It is strongly recommended that you choose this option before confirming the find / replace action.

8. Mark the Find / Replace in Entire Registry checkbox, to find and replace all occurrences of the required value in the entire registry. Marking this checkbox ensures that the find / replace function is not restricted to a single tree or sub-tree.

NOTE If you do not mark this checkbox, only the selected sub-tree will be searched for the required value.

9. Click OK to begin the find / replace action.

   When the find / replace action is complete, you will see a confirmation dialog showing the search results and replacements made. This dialog will give you a complete list of all the replacements that were made, along with complete tree structure of the keys whose values were replaced.

To Update the Installation Key

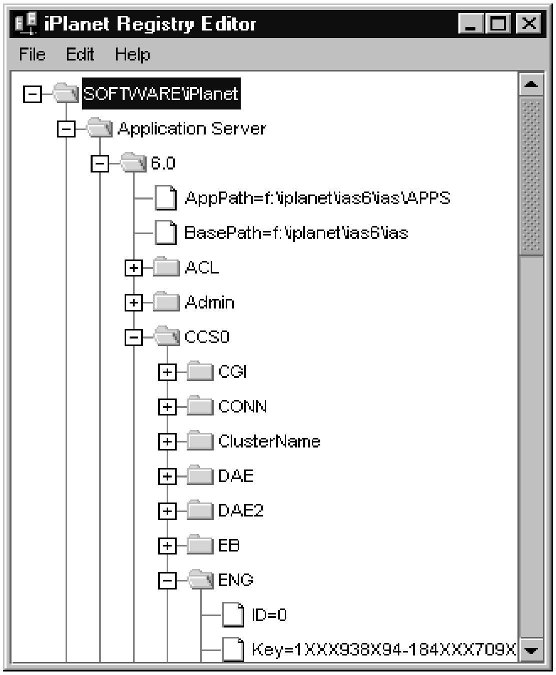
If you installed iPlanet Application Server with an evaluation license, the server stops running at the end of the evaluation period. You will need to update the installation key if you have extended the evaluation period or purchased the server. Updating the installation key saves you from having to reinstall the server software and reconfigure the environment.

To reset the installation key, perform the following tasks:

1. Shutdown iPlanet Application Server.
2. Open iPlanet Registry Editor.

   (See About iPlanet Registry Editor.)

   The following window appears:

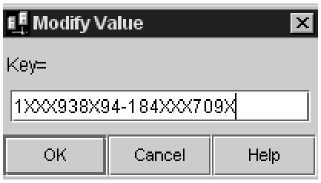


```
SOFTWARE\iPlanet
  Application Server
    6.0
      AppPath=f:\iplanet\ias6\ias\APPS
      BasePath=f:\iplanet\ias6\ias
      ACL
      Admin
      CCS0
        CGI
        CONN
        ClusterName
        DAE
        DAE2
        EB
        ENG
          ID=0
          Key=1XXX938X94-184XXX709X
```

3. Open the following key:

   `SOFTWARE\iPlanet\Application Server\6.0\CCS0\ENG`

4. Double-click the `Key` String value and enter the new Installation Key value.



Key=

1XXX938X94-184XXX709X

5. Click OK.

6. Close the registry editor.

7. Restart iPlanet Application Server.

Changing the IP Address

When the IP Address of a machine on which iPlanet Application Server has been installed changes, such as when the machine is moved or has been assigned to a different network, you need to replace the old IP Address with the new IP Address.

You need to replace the IP Address in the local iPlanet Registry of each instance of iPlanet Application Server. This is to let the network know the current IP Address of your machine.

If you have installed iPlanet Console (previously known as Netscape Console) on your machine, you need to let the other machines on the network know from where the iPlanet Console Administration Server can be accessed.

The procedure for changing the IP Address is described in the following sections:

- To Modify IP Address in the Local iPlanet Registry
- To Update IP Address of iPlanet Console Administration Server

NOTE If iPlanet Console Administration Server is not installed on your machine, you can ignore the second section.

To Modify IP Address in the Local iPlanet Registry

You can update the new IP Address in iPlanet registry using iPlanet Registry Editor, kregedit. When you do this, the IP Address is updated in both the local registry and in the Directory Server that stores your machine's configuration.

NOTE Multiple instances of iPlanet Application Server are supported on Solaris machines. If you have installed multiple instances of iPlanet Application Server, you need to replace the old IP Address with the new IP Address in the iPlanet Registry of each instance.

To change the IP address in iPlanet registry, perform the following tasks:

1. Start iPlanet Registry Editor

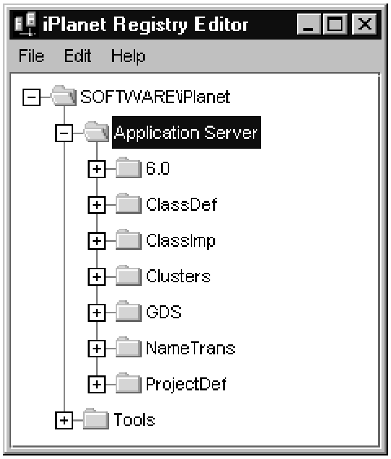
   (See About iPlanet Registry Editor.)

2. Select the following key:

   `SOFTWARE\iPlanet`

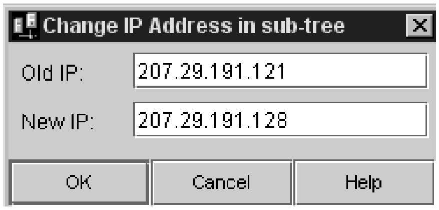
   This is the root key. You need to select this key, to update all occurrences of the old IP Address, in the entire registry

.



3. From the Edit menu, choose Change IP Address in sub-tree.

   The Change IP Address in sub-tree dialog box appears:



Old IP: 207.29.191.121

New IP: 207.29.191.128

4. In the Old IP text field, enter the old IP Address.
5. In the New IP text field, enter the new IP Address.

NOTE The IP Address you enter must be of this format: `129.29.191.128`. Each sequence of digits must be separated with a dot.

---

6. Click OK to save your changes.

NOTE

- If you enter the IP address along with the Port number (for example, `129.29.191.128:10818`), in the Change IP Address in sub-tree dialog box, only the IP Address will be extracted, validated and replaced in iPlanet Registry.
- To ensure that all occurrences of the old IP Address have been replaced in the registry, use the `Find/Replace` option from the Edit menu. Note that though this option checks for the correctness of the IP Address format, it does not validate the IP Address before replacing it in iPlanet Registry.

To Update IP Address of iPlanet Console Administration Server

iPlanet Console (previously known as Netscape Console) performs common server administration functions such as stopping and starting servers, installing new server instances, and managing user and group information through the LDAP services of Directory Server. Using iPlanet Console, you can manage remote instances of Directory Server that may exist on your network.

iPlanet Console comes with its own Administration Server, which is used internally by iPlanet Console. If iPlanet Console is installed on your machine, you need to ensure that all the machines in your network know the correct `nsserveraddress` (iPlanet Console Administration Server's IP Address) and the `nsadminaccessaddress` (the IP Address of the machine from which the iPlanet Console Administration Server can be accessed).

The `nsserveraddress` for the iPlanet Console Administration server is stored in two locations; Directory Server and the local configuration file of iPlanet Console.

The `nsadminaccessaddress` is stored only in the local configuration file of iPlanet Console.

The following sections describe how you can change the `nsserveraddress` and the `nsadminaccessaddress`:

- To Modify IP Address in Directory Server
- To Modify IP Address in Local Configuration File

To Modify IP Address in Directory Server

To change the `nsserveraddress` in Directory Server, perform the following tasks:

1. Go to `<iASInstallDir>/slapd-<hostname>` and run the following script:

   `db2ldif`

   This script generates an LDIF (LDAP Data Interchangeable Format) file, whose preliminary filename will bear the stamp of the date and time it was generated, in the `YYYY-MM-DD` format. For example, an LDIF file generated on 3rd June 2001 at 14 hours, 2 minutes and 38 seconds will look like this:

   `2001_06_03_140238.ldif`.

2. Go to `<iASInstallDir>/slapd-<hostname>/ldif` and locate the LDIF file with the latest date and time stamp.

3. Open the LDIF file (using any text editor) and search for the `nsserveraddress` entry. Replace the old IP Address with the new one. Note that you will need to search for all occurrences of the old IP Address and replace with the new one, to ensure that the change is complete.

4. Stop the Directory Server, by performing the following tasks:

   On Windows:

   a. From the Start menu, choose `Settings>Control Panel`.

   b. Double-click the Services icon

   c. In the Services window, select Netscape Directory Server.

   d. Click Stop to stop the server.

   On Solaris:

   a. Navigate to the `<iASInstallDir>/slapd-<hostname>` path.

   b. Run the following script:

      `stop-slapd`

      The Directory Server stops.

5. The next step is to write the LDIF file to the Directory Server. To do this, go to `<iASInstallDir>/slapd-<hostname>` and run the following script:

   `ldif2db -i ldif/<path>/<filename>.ldif`

   When you run this script, the changes made to the file are recorded in Directory Server.

NOTE You need to provide the absolute path of the LDIF file in which you changed the `nsserveraddress` value.

---

6. Start the Directory Server.

   To start Directory Server, follow the procedure described in Step 4 above. For Windows, click Start in the Services window to start the server. For Solaris, run the script `start-slapd` to start the server.

NOTE Check the log files that are stored in the following path, to make sure that all occurrences of the old IP Address have been replaced in the Directory Server:

```
<iASInstallDir>/slapd-<hostname>/logs
```

To Modify IP Address in Local Configuration File

To change the IP Address in the local configuration file(s) of the iPlanet Console Administration Server, perform the following tasks:

1. Go to `<iASInstallDir>/admin-serv/config` and open the `local.conf` file, using any text editor.
2. Locate `nsserveraddress` entries and change the IP Address value in them. The entries will look like this: `configuration.nsserveraddress.129.158.228.63`.
3. Locate `nsadminaccessaddress` entries if any and change the IP Address value. The entries will look like this: `configuration.nsadminaccessaddress.129.158.228.63`.
4. Stop and start the iPlanet Console Administration Server, by performing the following tasks

   On Windows:

   a. Go to `<iASInstallDir>/` and double-click the `stop-admin` executable to stop the server.

   b. Double-click the `start-admin` executable to start the server.

   On Solaris

   a. Go to `<iASInstallDir/>` and run the `stop-admin` script to start the server.

   b. To start the server, run the `start-admin` script from the same path.

NOTE

- If multiple instances of iPlanet Application Server have been installed on a single host (supported only on Solaris machines), a local configuration file is present for each instance. When the IP Address changes, you need to update the required entries in the local configuration file for each instance of iPlanet Application Server.
- If you have installed multiple instances of iPlanet Application Server on your machine, you need to stop and start the Netscape Administration Server for each instance, for the changes to take effect.

To Configure Support for Multiple LDAP Domains

iPlanet Application Server supports multiple LDAP domain names, based on the Nortel LDAP schema.

A user can now use different domain names such as `iPlanet.com`, `sun.com`, `netscape.com` to log on to an application. Each domain name that a user is associated with is entered into Directory Server.

To use multiple domains, you need to add the new domain name(s) to iPlanet registry and create the corresponding backend entries in the Directory Server. This can be done on both Solaris and Windows.

This section includes the following topics:

- To Add a Domain Name to iPlanet Registry.
- To Configure the New Domain Name in Directory Server

To Add a Domain Name to iPlanet Registry.

To add a domain name to iPlanet registry, perform the following tasks:

1. Open the iPlanet Registry Editor.

(See About iPlanet Registry Editor.)

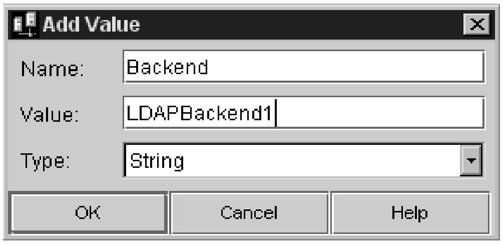
2. Open the following key:

```
Software/iPlanet/Application Server/6.0/Principal
```

3. Create a subkey using a new domain name, for example, *`iPlanet.com`*.

4. Select the new domain name and from the Edit menu, choose `Edit>Add Value`.

   The Add Value dialog box appears, as shown in the figure:

   Name: Backend

   Value: LDAPBackend1

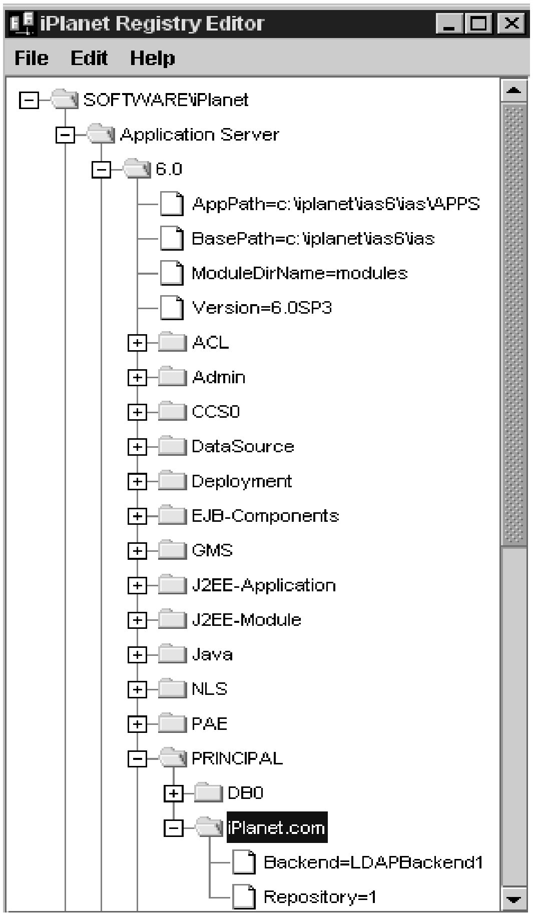
   Type: String

5. In the Name text field, provide a name for the value, for example, `Backend`.
6. In the Value field, specify a value, for example, `LDAPBackend1`. The value you specify here must find a corresponding value of the same name in Directory Server.

   For more information, refer to To Configure the New Domain Name in Directory Server.
7. Click OK to register the value in iPlanet Registry.

   You now need to create a second value for the domain subkey, as described in the following steps.
8. Select the new domain key, and choose `Add Value` from the Edit menu.
9. In the Add Value dialog box, provide a name, for example, `Repository`.
10. Specify a value, for example, `1`. Select `Integer` from the Type drop-down list.

You have now defined a new domain name in iPlanet Registry. When you complete the process of defining a new domain name in iPlanet Registry, your entries must look like the following example:



```
SOFTWARE\iPlanet
    Application Server
        6.0
            AppPath=c:\iplanet\ias6\ias\APPS
            BasePath=c:\iplanet\ias6\ias
            ModuleDirName=modules
            Version=6.0SP3
            ACL
            Admin
            CCS0
            DataSource
            Deployment
            EJB-Components
            GMS
            J2EE-Application
            J2EE-Module
            Java
            NLS
            PAE
            PRINCIPAL
                DB0
                iPlanet.com
                    Backend=LDAPBackend1
                    Repository=1
```

To Configure the New Domain Name in Directory Server

After you define the new domain name in iPlanet Registry, you need to configure the domain name in Directory Server. The domain name entries in iPlanet Registry will point to the entries that you create in Directory Server.

To configure the new domain name in Directory Server, perform the following tasks:

1. In iPlanet registry, open the key `Software/iPlanet/Application Server/GDS/Backends/LDAP/`

2. Create a subkey with the same value that you specified under `Software/iPlanet/Application Server/6.0/Principal`, i.e, `LDAPBackend1`.
3. Create a value for this subkey, which defines the attributes of the Common Name (cn) that is defined in Directory Server. For example: `- root = cn =global, cn=iasconfig, cn=iAScluster, O=NetscapeRoot`.
4. Create a new subkey under the key `LDAPBackend1`, for example `0`.
5. Create the following values for the subkey `0`.
   - `GroupPath = ou=Groups, o=<new domain>`
   - `Host = <hostname>`
   - `Password = <encrypted password>.` See note below.
   - `Port = <Directory Server Port>.` The default port is `389`.
   - `User =` *`cn=Directory Manager`*
   - `UserPath =` *`ou=People, O=<new domain>`*

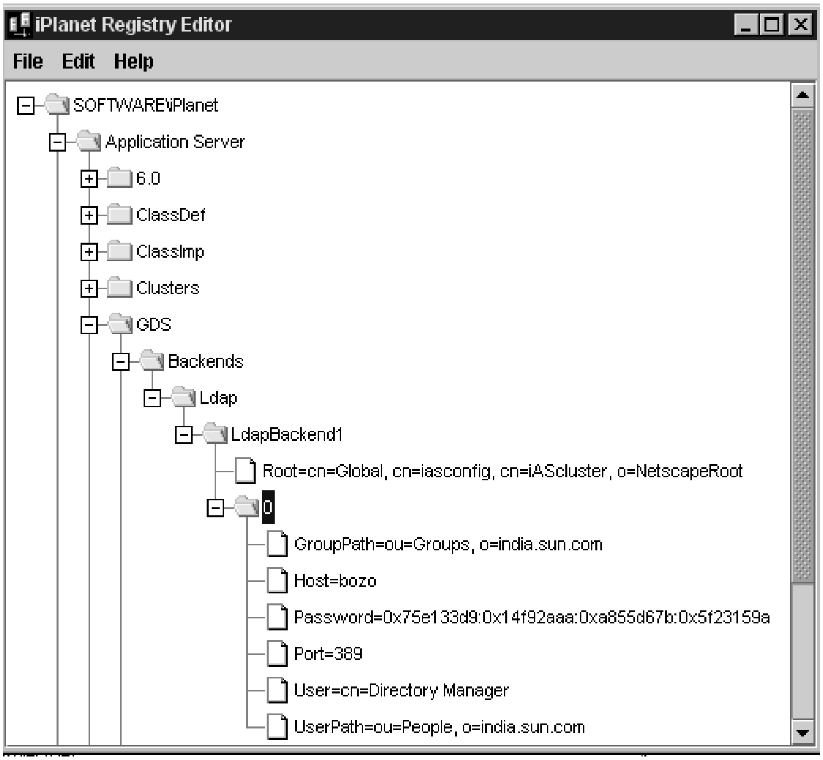
This procedure configures the back-end entries for the new domain name in the Directory Server.

NOTE You need an encrypted password when you login using your new domain name, through which you are identified by Directory Server. You can generate an encrypted password by running the following script:

```
<iASInstallDir>/ias/bin/kencrypt2 <text to be encrypted>
```

This script creates an encrypted password out of the alpha-numeric string that you provide. Your password can comprise as many characters that you want.

When you finish creating the required back-end entries for the new domain name, your entries in iPlanet Registry must look like the following example:



```
SOFTWARE\iPlanet
  Application Server
    6.0
    ClassDef
    ClassImp
    Clusters
    GDS
      Backends
        Ldap
          LdapBackend1
            Root=cn=Global, cn=iasconfig, cn=iAScluster, o=NetscapeRoot
            0
              GroupPath=ou=Groups, o=india.sun.com
              Host=bozo
              Password=0x75e133d9:0x14f92aaa:0xa855d67b:0x5f23159a
              Port=389
              User=cn=Directory Manager
              UserPath=ou=People, o=india.sun.com
```

Part 2

Administering a Single iPlanet Application Server

Chapter 2

Monitoring Server Activity

This chapter describes the monitoring service provided by iASAT. This service allows you to chart various attributes of the Executive, Java, C++ and Bridge server processes.

The following topics are included in this chapter:

- Monitoring iPlanet Application Server
- Receiving Event Notification

Monitoring iPlanet Application Server

iPlanet Application Server Administration Tool (iASAT) provides a monitoring service that lets you chart the activity of the Executive, Java, C++ and Bridge servers that make up iPlanet Application Server. You can also log the information to a file. By graphically representing this server activity or recording the data in a file, you can track and review the performance of an application server or group of servers and make adjustments to improve performance. For example, if you add more memory to the application server or deploy a new application, you may want to monitor the performance of the application server to see the impact of these changes.

iPlanet Application Server's monitoring service polls the application server at designated intervals. This saves server resources because the server updates the information being monitored at the specified interval instead of updating it continuously. You can specify this interval in the Monitoring window. For information about setting the interval time, see To Change a Process Data Plot.

The monitoring window "pops out" from the Administration Tool when you click a process to monitor. This detached window enables you to monitor server activity in a separate window while continuing to perform other administrative tasks using the Administration Tool.

This section describes the following topics:

- Monitoring Process Attributes
- To Log Process Data to a File
- To Change a Process Data Plot
- To Remove a Process Data Plot

Monitoring Process Attributes

The server activity, or attributes, you can chart varies according to which server, or process, you are monitoring.

The Executive Server (KXS) process is responsible for managing and hosting the system-level services, such as the load-balancing service, and for delegating requests to one of the application processes, either the Java server, or C++ server depending on the language in which the application component is written.

You can chart the following attributes of the Executive Server process:

Table 2-1 Executive Server Monitoring Attributes

Executive Server Process Attribute (KXS)	Description
CPU load	The amount of load on the CPU on which this Executive Server process is running, as calculated by the load balancing service.
Disk input and output	The rate of Read and Write operations issued by the system on which this Executive Server is running, as calculated by the load balancing service.
Memory thrash	The number of pages read from or written to the hard disk drive to resolve memory references to pages that were not in memory at the time of the reference.
Requests queued	Number of requests currently waiting in the queue for processing.
Cached results	Number of entries stored in the result cache.

Table 2-1 Executive Server Monitoring Attributes *(Continued)*

Executive Server Process Attribute (KXS)	Description
Average execution time	Average amount of time for the Executive Server process to execute a request.
Requests/interval	Number of new requests received since the last polling.
Total requests	Total number of requests the process has received starting up (This value is reset to 0 upon server or process start-up.). For the executive process, this corresponds to the total number of requests the server has executed across all server processes.
Current requests	The number of requests currently being processed by the server; includes all requests dispatched and being processed in the KJS/KCS engines.
Requests waiting	Number of queued requests waiting to be serviced.
Requests ready	Number of queued requests ready to be serviced.
Current Requests Threads	Number of request threads allocated by the process (includes both idle threads and threads actively processing requests). Note that this number cannot exceed Maximum Threads, or fall below Minimum Threads configured for this process. These values are set in the General window.
Requests Threads Waiting	Number of requests threads available to execute new incoming requests. This number will be a subset of the Current Requests Threads monitoring attribute.
Total Threads	Number of threads being used by the process.
Bytes sent/interval	Number of new bytes sent since the last polling.
Bytes received/interval	Number of new bytes received since the last polling.
Current Sessions	Number of current sessions being handled.

NOTE If you monitor CPU load, disk input and output, or memory thrash, you must specify the intervals at which the statistics for these process attributes are updated. To set the intervals, select Load Balancing. Choose User Defined Criteria Load Balancing, then click the Advanced Settings tab.

The Java Server (KJS) and C++ Server (KCS) processes are responsible for hosting application elements, depending on the language in which the element is written. The Java Server hosts application components written in Java, and the C++ Server hosts components written in C++. In addition, the Corba Executive Server (CXS) or Bridge process allows for independent Java clients (Rich Clients) to communicate directly to Enterprise JavaBeans hosted on a Java Server. For more information about Rich Client, see the *Developer's Guide (Java)*.

You can chart the following attributes of the Java, C++ and Bridge Server processes:

Table 2-2 Java, C++ and IIOP Bridge Server Monitoring Attributes

Java/C++ and Bridge Server Processes (KJS, KCS and CXS) Monitoring Attributes	Description
Average execution time	Average amount of time for the process to execute a request.
Requests/interval	Number of new requests received since within the interval.
Total requests	Total number of requests the process has received since the last start-up. This value is reset to zero upon server or process start-up.
Current Requests	The number of requests currently being processed by this process.
Requests Waiting	Number of queued requests waiting to be serviced.
Requests Ready	Number of queued requests ready to be serviced.
Current Requests Threads	Number of request threads allocated by the process (includes both idle threads and threads actively processing requests). Note that this number cannot exceed Maximum Threads, or fall below Minimum Threads configured for this process. These values are set in the General window.
Request Threads Waiting	Number of requests threads available to execute new incoming requests. This number will be a subset of the Current Requests Threads monitoring attribute.
Active data connections	Number of currently active data connections.
Cached data connections	Number of currently cached data connections.
Queries/interval	Number of queries executed within the interval.
Trans committed/interval	Number of transactions committed within the interval.

Table 2-2 Java, C++ and IIOP Bridge Server Monitoring Attributes *(Continued)*

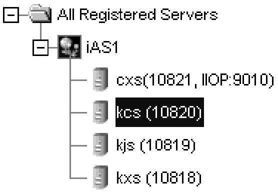
Java/C++ and Bridge Server Processes (KJS, KCS and CXS) Monitoring Attributes	Description
Trans rolledback/interval	Number of transactions rolled back within the interval.
Total Threads	Number of threads being used by the process.
Bytes sent/interval	Number of new bytes sent since the last polling.
Bytes received/interval	Number of new bytes received since the last polling.

For each process, you can chart one or more attributes. You can also simultaneously chart the attributes of several application servers, if you have a multiple-server enterprise.

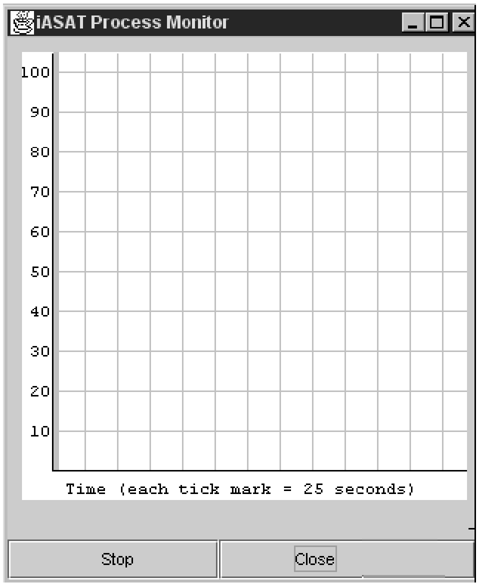
To Monitor Process Attributes

To monitor process attributes, perform the following tasks:

1. On the iASAT toolbar, click Monitor to open the Monitor window.
2. In the left pane of the Monitor window, click the process whose attributes you want to chart, as shown in the following figure:

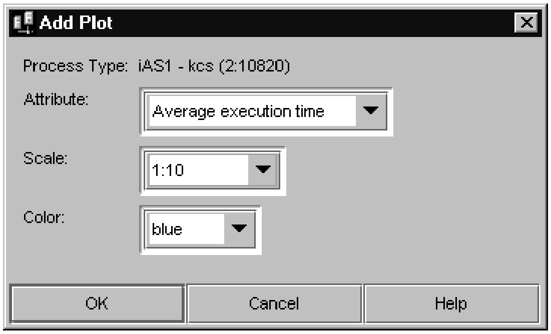


A separate monitoring panel pops out of iASAT, as shown in the following figure:



3. In the right pane of the monitoring window in iASAT, click Add Plot located at the bottom of the window.

   The Add Plot dialog opens for you to specify the attributes to monitor for the highlighted process.



Process Type: iAS1 - kcs (2:10820)

Attribute: Average execution time

Scale: 1:10

Color: blue

4. In the Attribute drop-down list, select the attribute to chart.

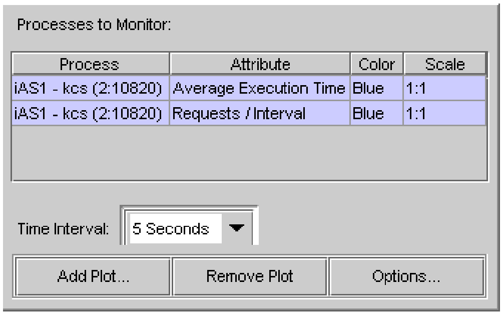
5. From the Scale drop-down list, choose the ratio (scale) at which to plot the attribute.

   Values range from `10:1` to `1:1,000,000`. A scale of `10` to `1 (10:1)` indicates that 10 units will be plotted on the Process Monitor window for each attribute count.

6. From the Color drop-down list, choose a color to represent the process attribute on the chart.

7. Repeat steps 2 through 6 for each process or attribute you want to chart.

   Each process attribute that you choose to chart is displayed in the Monitor window.



Processes to Monitor:

Process	Attribute	Color	Scale
iAS1 - kcs (2:10820)	Average Execution Time	Blue	1:1
iAS1 - kcs (2:10820)	Requests / Interval	Blue	1:1

Time Interval: 5 Seconds

8. In the Time Interval drop-down list, select the interval at which you want to update the Monitor Plot window.

   This setting applies to all process attributes displayed in the Monitor window.

To Log Process Data to a File

Once you begin monitoring a process attribute, you can send data collected by the monitoring service to a file.

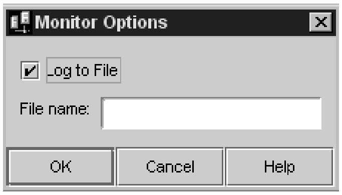
To log process data to a file, perform the following steps:

1. On the iASAT toolbar, click Monitor to open the Monitor window.

2. In the left pane of the Monitor window, click the process whose data you want to chart.

3. In the right pane of the Monitor window, click Options.

The following dialog box appears:



☑ Log to File

File name:

4. Click the Log to File checkbox to enable the logging service.
5. In the File Name text field, enter the name of the file where data is to be written.
6. Click OK to save your changes.

To Change a Process Data Plot

Once an attribute data plot is specified for a process (KCS, KJS, and KXS), you can adjust the plot using the Attribute, Color, and Scale drop-down boxes.

To change the way a process attribute is plotted, perform the following steps:

1. On the iASAT toolbar, click Monitor to open the Monitor window.
2. In the right pane of the window, select a process row, whose attributes you want to change.
3. To change an attribute, click the Attribute column and choose a new attribute to plot from the drop-down list.
4. Similarly, click the Color and Scale columns to change how the attribute will be plotted.

To Remove a Process Data Plot

If you no longer want to plot an attribute for a process, you can remove it from the plot.

To remove a process plot, perform the following steps:

1. Click Monitor on the iASAT toolbar to open the Monitor window.

2. In the left pane of the Monitor window, choose a process, to display the process rows in the right pane.
3. In the right pane of the Monitor window, select a process whose attribute you want to remove.
4. Click Remove Plot.

   The attribute is removed from the Monitor window.

Receiving Event Notification

Event notification is useful when you cannot actively monitor an iPlanet Application Server. This passive monitoring system is activated only in critical circumstances, such as when a process has failed.

You can set the system to alert one or more concerned parties through email when a critical situation arises by supplying the email addresses of those you want to alert. In addition, you can specify a script that will run automatically when certain events occur.

This section includes the following topics:

- About Events
- To Configure Email Notification for an Event
- To Specify an Event-Invoked Script

About Events

You can specify an individual to notify or a script to run for the following critical events:

- Executive Sever (KXS) goes down
- Java Server (KJS) goes down
- C++ Server (KCS) goes down
- Process auto restarts exceeded
- Abnormal Cluster is detected

This section includes details for the following topics:

- What Do I Do When a Server Goes Down?
- What Do I Do When Restarts Are Exceeded?
- What Do I Do When an Abnormal Cluster is Detected?

What Do I Do When a Server Goes Down?

If one or more of the Executive Server, Java Server, or C++ Server processes go down, the Administrative Server attempts to restart each process. If the process cannot be restarted by the Administrative Server, the application stops running and can result in lost transactions.

Recurring failures are usually attributed to problems within the application code, but other failures can also happen. Regardless of what causes a process to fail, it is useful to be notified immediately.

If the process restarts, investigate the cause of the failure to determine whether adjustments can be made to prevent future failures. If the process does not restart, look at the log to find the cause of the failure.

What Do I Do When Restarts Are Exceeded?

You can also be notified when the Administrative Server has exceeded the number of times it has been set to restart a process. The maximum engine restarts value is set on the Server tab of the General window.

Increase the Administrative Server restart option, if it is low, and determine the cause of the process failure.

What Do I Do When an Abnormal Cluster is Detected?

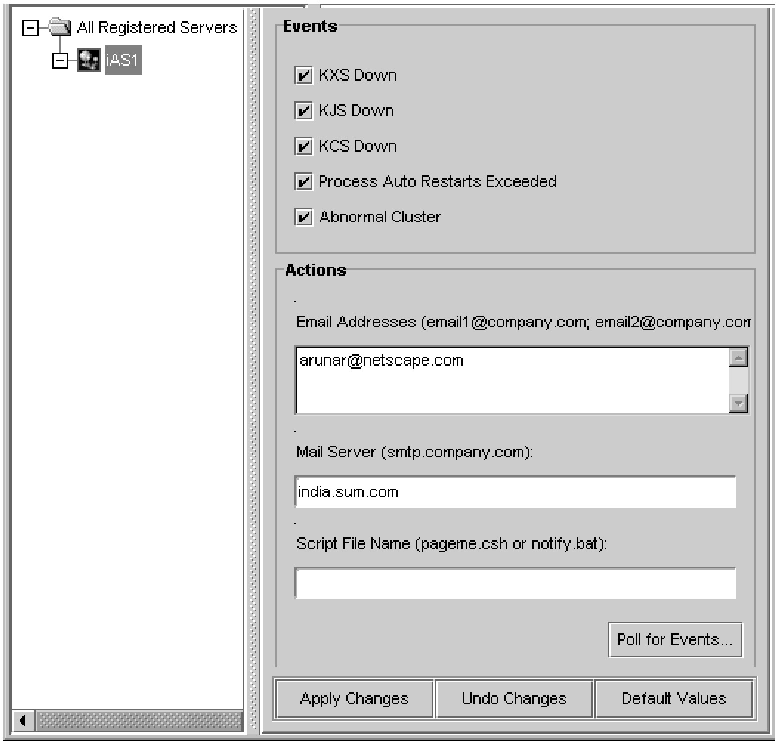
You can also be notified when an abnormal cluster condition has been detected. Within a normal operating cluster there is one sync primary iPlanet Application Server that is the primary data store, with which all other cluster members communicate for the latest distributed data information. An abnormal cluster is where a dual-primary or a no-primary condition has been detected.

Mark the Restart in case of abnormal cluster checkbox on the Cluster tab of the General window. iPlanet Application Server will re-start an appropriate process so that one (and only one) sync primary is present in the cluster. For more information about clusters, see Managing Distributed Data Synchronization.

To Configure Email Notification for an Event

To send an email notification for an event, perform the following steps:

1. On the iASAT toolbar, click Events to open the Events window.
2. From the left pane of the Events window, select the server for which you want to configure events.
3. From the right pane of the Events window, select the event or events for which you want to be notified by clicking the corresponding checkbox as shown in the following figure:



Events

☑ KXS Down

☑ KJS Down

☑ KCS Down

☑ Process Auto Restarts Exceeded

☑ Abnormal Cluster

Actions

Email Addresses (email1@company.com; email2@company.con

arunar@netscape.com

Mail Server (smtp.company.com):

india.sum.com

Script File Name (pageme.csh or notify.bat):

4. In the Email Addresses field, specify the email address or addresses of the persons you want to send notification. To use multiple email addresses, separate each email address with a semi-colon(`;`), as shown in the following example:

```
betsy@doghouse.com;arland@meow.com
```

5. In the Mail Server field, specify the mail server through which the notification is sent. Use the following format:

   ```
   mail.company.com
   ```

6. To see the most recent events that might have been sent out for this server, click Poll for Events.

   The Poll for Event dialog box appears displaying a list of the recent events for the selected server.

NOTE When you click Poll for Events, events are consumed (that is, the events that you see are no longer included in the next set of events that are displayed).

7. Click Apply Changes to save your changes.

To Specify an Event-Invoked Script

You can configure the event notification service to run a script. The script might page the system administrator, bringing the problem to the administrator's attention, or perform any other automated task that will help keep the system running smoothly when faced with a critical event.

When a script runs, it passes an argument to indicate what type of event has occurred. For instance, the following command indicates that a Java Server (KJS) process has crashed:

```
/script location/ crash kjs
```

To configure the event notification service to run a script in response to an event, perform the following steps:

1. On the iASAT toolbar, click Events to open the Events window.
2. From the left pane of the Events window, select the server for which you want to configure events.
3. In the right pane of the Events window, mark the checkboxes against the events for which you want to invoke a script.
4. In the Script field, specify the path of the script to run. For example:

   ```
   /mydir/scripts/myscript.pl
   ```

5. Click Apply Changes to save your changes.

Chapter 3

Configuring SNMP to Monitor iPlanet Application Server with Third-Party Tools

This chapter describes how to configure Simple Network Management Protocol (SNMP) so that you can monitor iPlanet Application Server with third-party SNMP management tools.

The following topics are included in this chapter:

- About SNMP
- Enabling SNMP Statistics Collection
- About the Management Information Base (MIB)
- Setting Up the Master Agent and SubAgents

About SNMP

SNMP is a protocol used to exchange data about network activity. With SNMP, data travels between your application server and a workstation where network management software is installed. From this workstation, you can remotely monitor your network and exchange information about network activity between servers. For example, using an application like HP OpenView, you can monitor which iPlanet Application Server machines are running, as well as the number and type of error messages your application servers receive.

Your network management workstation exchanges information with the application servers in your enterprise through two types of agents: the subagent and the master agent. The subagent gathers information about an application server and passes that information to the master agent. The master agent exchanges information between the various subagents and the network management workstation. The master agent runs on the same host machine as the subagents with which it communicates.

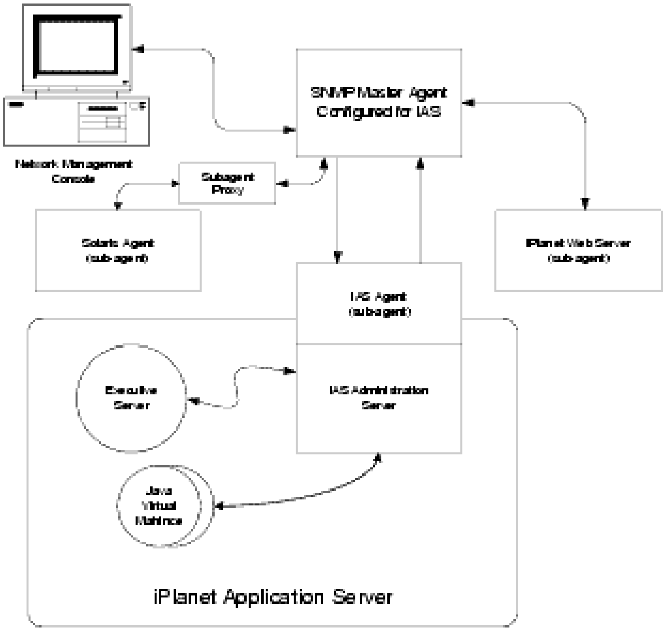

SNMP Master Agent Configured for IAS
Network Management Console
Subagent Proxy
Solaris Agent (sub-agent)
iPlanet Web Server (sub-agent)
IAS Agent (subagent)
Executive Server
IAS Administration Server
Java Virtual Mahince
iPlanet Application Server

Figure 3-1 SNMP Agent Support Architecture

Enabling SNMP Statistics Collection

The iPlanet Application Server SNMP subagent does not report SNMP statistics to the network management workstation unless you enable statistics collection. If statistics collection is not enabled, the subagent cannot be started.

NOTE If the network management workstation experiences difficulty obtaining SNMP statistics, check the server log information located at the following path:

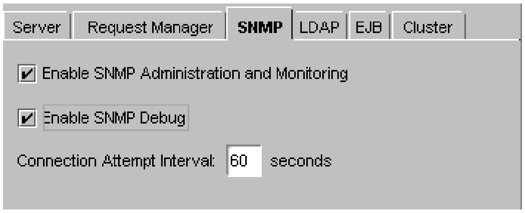
```
<iASInstallDir>/<mail-instanceName>/log/default/
```

If the SNMP data collection process (`snmpcoll`), is not running, check the iPlanet Administration Console to see whether the SNMP enable flag is on. For more information, see *Managing Servers with Netscape Console* at `http://docs.iplanet.com/`

If you disable the startup server, this collection process is also disabled.

To enable data collection, perform the following steps:

1. Click General on the iPlanet Application Server Administration Tool (iASAT) toolbar to open the General window.
2. In the right pane of the General window, click the SNMP tab.



☑ Enable SNMP Administration and Monitoring

☑ Enable SNMP Debug

Connection Attempt Interval: 60 seconds

3. Mark the Enable SNMP Administration and Monitoring check box.

   This step enables the SNMP subagent to publish statistics about the application server to the master agent.
4. Mark the Enable SNMP Debug checkbox, to log error messages if there is a problem with connecting to the master agent.
5. Specify an interval in the Connection Attempt interval text field.

This is the time interval in which the subagent will attempt to connect to the master agent.

NOTE You will have to re-start the iPlanet Application Server server for these settings to take affect.

About the Management Information Base (MIB)

iPlanet Application Server stores variables pertaining to network management in a tree-like hierarchy known as the server's management information base (MIB). iPlanet Application Server reports significant events to the network management workstation by sending messages containing these variables. The network management workstation can also query the server's MIB for data or can remotely change variables stored in the MIB.

This section includes the following topics:

- Formatting MIB Entries
- Making MIB Available on SNMP Third-Party Management Software

Formatting MIB Entries

The MIB file contains the definitions for managed objects, or variables, that store network information for the server. Each variable definition includes the variable name, its data type and read / write access level, a brief description, and a permanent object identifier.

This sample entry shows the definition for the nsmailEntityDescr variable:

```
nasKesMaxThread OBJECT-TYPE     / object type

SYNTAX      INTEGER (SIZE (1..512))      / syntax

ACCESS      read-write      / read/write access
level

STATUS      mandatory      / status

DESCRIPTION                / description
"The maximum number of threads used to serve requests."

::= { kes 4 }  / object identifier
```

This definition contains the following information:

- Object Type: gives the name of the variable, in this case, `iasKesMaxThread`.
- Syntax: gives the abstract data type of the variable object type in ASN.1 notation. For example, the Syntax of the `nasKesMaxThread` variable is `INTEGER (SIZE (1..512))`.
- Access: gives the required access level to the variable. Possible access levels are read-only, read-write, write-only, or not-accessible.
- Status: tells whether the element is mandatory, optional, or obsolete.
- Description: text description of the element, enclosed in quotes. For example, the description of the `nasKesMaxThread` variable is "The maximum number of threads used to serve requests."
- Object Identifier: assigned name that serves as a permanent identifier for each managed object in the MIB name tree in its name space. Objects in SNMP are hierarchical; the object identifier is a sequence of labels that represents the object in the hierarchy. For example, nasKesMaxThread is identified as `kes 4.` This means that it has the label 4 in the subtree kes.

  Note that `kes`, in turn, has the label `4` in the `kesTable subtree`.

Making MIB Available on SNMP Third-Party Management Software

Refer to the SNMP management software for detailed procedures for making the MIB available. In general, you have to copy the iPlanet Application Server MIB to the Network Management machine and then load it into the SNMP management software's MIB database.

You can find the iPlanet Application Server MIB database in the following location:

```
<iASInstallDir>/plugins/snmp/
```

Setting Up the Master Agent and SubAgents

The SNMP Master agent is native to your Solaris operating system. Master agent operation is defined in an agent configuration file called `CONFIG`. You can edit the `CONFIG` file manually.

NOTE This procedure assumes that you are running Solaris 2.6 with recommended patches. It also assumes that iPlanet Web Server is installed.

This section includes the following topics:

- To Configure the Master SNMP Agent
- To Start the SNMP Master Agent
- To Verify SNMP Configuration

To Configure the Master SNMP Agent

To configure the master SNMP agent, perform the following steps:

1. Log in as root.
2. Check to see if there is a Solaris SNMP daemon (`snmpdx`) running on port 161.

   If an SNMP daemon is running, make sure you know how to restart it and which MIB trees it supports. Then kill its process.
3. Edit the Solaris SNMP daemon start-up file `s76snmpdx` in `/etc/rc3.d` to modify the port to which the daemon listens.

   In the start section, replace the line

   ```
   /usr/lib/snmp/snmpdx -y -c /etc/snmp/conf
   ```

   or

   ```
   /usr/lib/snmpdx -p 161 -y -c /etc/snmpconf
   ```

   with

   ```
   /usr/lib/snmp/snmpdx -p 1161 -y -c /etc/snmp/conf
   ```

   You have changed the port to which the daemon listens from `161` to `1161`.

4. Edit the `CONFIG` file located in `<*iASInstallDir*>/ias/snmp` in the server root directory.

   The `CONFIG` file defines the community and the manager that the master agent will work with. The manager value should be a valid system name or an IP address. The following is an example of a basic `CONFIG` file:

```
COMMUNITY      public
               ALLOW ALL OPERATIONS

MANAGER        your_manager_station_name
               SEND ALL TRAPS TO PORT 162
               WITH COMMUNITY public
```

5. (Optional) Define `sysContact` and `SysLocation` variables in the `CONFIG` file.

   You can edit the `CONFIG` file to add initial values for `sysContact` and `sysLocation` which specify the `sysContact` and `sysLocation` MIB-II variables. Note that the strings for `sysContact` and `sysLocation` in this example are enclosed in quotes. Any string that contains spaces, line breaks, tabs, and so on must be in quotes. You can also specify the value in hexadecimal notation.

   In this sample CONFIG file, `sysContract` and `sysLocation` variables are defined:

```
COMMUNITY     public
              ALLOW ALL OPERATIONS

MANAGER       nms2
              SEND ALL TRAPS TO PORT 162
              WITH COMMUNITY public

INITIAL       sysLocation "Server room 501 East Middlefield
              Road Mountain View, CA 94043 USA"

INITIAL       sysContact "John Doe email: <jdoe@iPlanet.com>"
```

The encapsulator forwards requests from the master agent to the Solaris agent that now listens on port 1161.

6. Edit the file CONFIG_SAGT, modifying the following lines:

   ```
   Agent at 1161 with Community Public
   ```

   This configures the subagent to serve the Solaris agent on port 1161.

   ```
   Subtrees <list of oids>
   ```

   This configures the SNMP subtrees served by the Solaris agent.

   ```
   Forward All Traps
   ```

   This ensures that all traps sent by the Solaris agent are forwarded to the master agent.

To Start the SNMP Master Agent

Once you have installed the SNMP master agent, you can start it manually or by using iPlanet Console.

To start the master agent manually, enter the following at the command prompt:

```
# magt CONFIG INIT &
```

The `INIT` file is a nonvolatile file that contains information from the MIB-II system group, including system location and contact information. If `INIT` doesn't already exist, starting the master agent for the first time will create it. An invalid manager name in the `CONFIG` file will cause the master agent start up to fail.

NOTE	`INIT` contains information about the local system. This file is created the first time you start the master agent. You should not copy this file across machines.

To automatically start the master agent when you start the server, perform the following steps:

1. Edit the files `ias/snmp/k75snmpmagt` and `ias/snmp/S75snmpmagt`.
2. Change `$GX_ROOTDIR` to the iPlanet Application Server installation directory path if this variable is not yet defined in the root’s environment.
3. Copy `k75snmpmagt to /etc/rc2.d` and `s75snmpmagt to /etc/rc3.d`.

To start a master agent manually on a nonstandard port, use one of two methods:

- Method 1: In the `CONFIG` file, specify a transport mapping for each interface over which the master agent listens for SNMP requests from managers. Transport mappings allow the master agent to accept connections at the standard port and at a nonstandard port. The master agent can also accept SNMP traffic at a nonstandard port. The maximum number of concurrent SNMP is limited by your target system's limits on the number of open sockets or file descriptors per process. The following is an example of a transport mapping entry:

```
TRANSPORT        extraordinary SNMP
                 OVER UDP SOCKET
                 AT PORT 11161
```

  After editing the `CONFIG` file manually, you should start the master agent manually by typing the following at the command prompt:

```
# magt CONFIG INIT&
```

- Method 2: Edit the `/etc/services` file to allow the master agent to accept connections at the standard port as well as at a nonstandard port.

To Verify SNMP Configuration

After you have performed the procedures outlined in this chapter, you can verify SNMP setup.

To verify SNMP, perform the following:

1. Stop iPlanet Application Server. For more information on starting and stopping iPlanet Application Server, see To Start and Stop a Server Using iASAT.

   Also make sure that all iPlanet Application Server processes (KAS, KXS, KJS and KCS) are stopped.

2. Verify that Directory Server is running. If it is not running start Directory Server, by executing the following command:

```
<iASInstallDir>/slapd-<hostname>/start-slapd
```

3. Verify that iPlanet Web Server (iWS) (e.g. https-servername) is running. If it is not running start it as follows:

   ```
   /usr/iplanet/suitespot/https-solsystem/start
   ```

   where `solsystem` is the server name.

4. Verify that the Solaris SNMP agent (snmpdx) is running using the UNIX ps command as follows:

   ```
   ps -ef | grep snmpd
   ```

   If it is not running, start it with:

   ```
   /etc/rc3.d/S76snmpclx start
   ```

5. Verify that the iPlanet Application Server Master Agent (magt) and encapsulator/proxy subagent (sagt) are running using the UNIX ps command as follows:

   ```
   ps -ef | grep magt
   ```

   ```
   ps -ef | grep sagt
   ```

   If they are not running, start them with:

   ```
   /etc/rc3.d/S75snmpmagt start
   ```

6. Start iPlanet Application Server.

7. Use your third-party SNMP management software's MIB browser or test utility (for example, `snmpwalk`) to confirm that SNMP data is being collected.

Chapter 4

Logging Server Messages

This chapter describes the message-logging service provided by iPlanet Application Server.

The following topics are included in this chapter:

- About the Logging Service
- About Web Server Requests
- About The Administration Server
- About DSync Logging Options
- About Monitoring iPlanet Application Server Log Files

About the Logging Service

You can enable the logging of server messages using the iPlanet Application Server message-logging service. The logging service is configured through the iPlanet Application Server Administration Tool Logging window. Here you can specify the destination and types of messages logged.

When you enable logging, iPlanet Application Server records messages generated by iPlanet Application Server application-level and system-level services. These messages describe the events that occur while a service is running. For example, each time iPlanet Application Server communicates with the database, the logging service records the resulting messages generated by database access service. Log files are stored in different servers. Logs pertaining to administration and deployment are stored in the Administration Server, known as KAS. The other servers that store message and event logs are KJS (Java Server), KCS (C++ Server) KXS (Executive Server). KAS starts the other three servers and monitors their activities.

This section includes the following topics:

- To Determine Types of Messages to Log
- Determining the Logging Destination

To Determine Types of Messages to Log

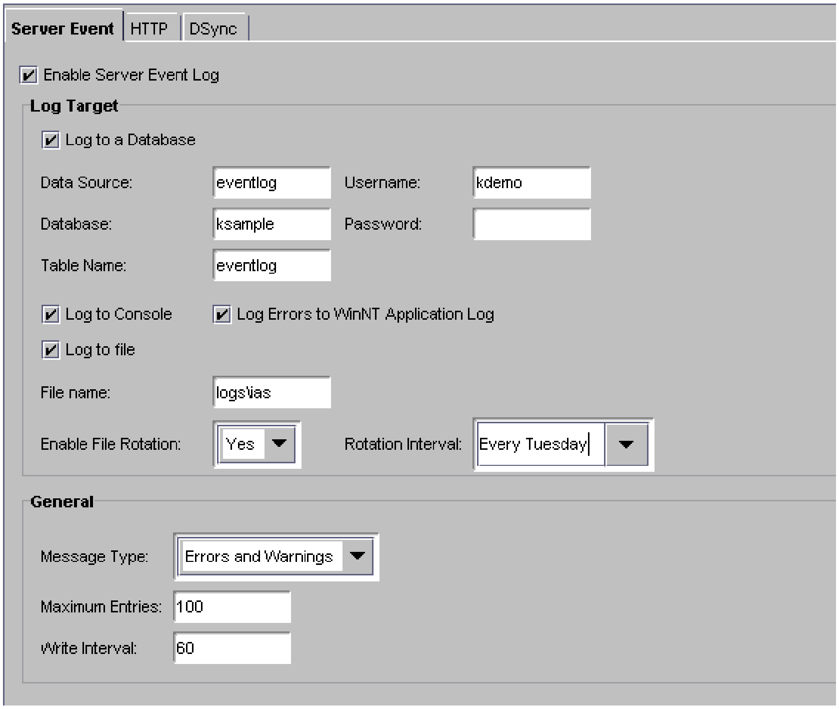
You can log any of the three types of messages generated by iPlanet Application Server services. Each type is described in the following table:

Table 4-1 Log Message Types

Message type	Description	When it might appear
Information message	Describes the processing of a request or normal service activity, such as a status update.	When no problems arise.
Warning message	Describes a noncritical problem that might be an indication of a larger problem.	When a service encounters a temporary problem, such as when it is unable to connect to a process.
Error message	Describes a critical failure of the service, from which recovery is not likely.	When a service encounters a critical problem, such as a pipe closure.

With the logging service, you can record error messages, error and warning messages, or all messages. To choose which type of messages to log, perform the following tasks:

1. Click Logging on the iPlanet Application Server Administration Tool (iASAT) toolbar to open the Logging window.
2. In the left pane of the Logging window, select the iPlanet Application Server node for which you want to specify log settings.
3. In the right pane of the Logging window, click the Server tab. The following dialog box appears:



☑ Enable Server Event Log

Log Target

☑ Log to a Database

Data Source: eventlog Username: kdemo

Database: ksample Password:

Table Name: eventlog

☑ Log to Console ☑ Log Errors to WinNT Application Log

☑ Log to file

File name: logs\ias

Enable File Rotation: Yes Rotation Interval: Every Tuesday

General

Message Type: Errors and Warnings

Maximum Entries: 100

Write Interval: 60

4. Mark the Enable Server Event Log checkbox, to enable event logging.
5. In the General pane, from the Message Type drop-down box, select one of these; Errors, Errors and Warnings, All Messages.
6. In the Maximum Entries text field, specify the maximum number of entries that can exist before data is written to the log.
7. In the Write Interval field, enter the amount of time (in seconds) that elapses before data is written to the log.

Logging Application Messages

Message logging is also useful for tracking and debugging application errors. By using the `log( )` method, application developers can send messages to the same log destination that the server administrator configures for iPlanet Application Server services.

For example, if an application encounters a problem in a segment of code, you can log the associated error message. Informational messages about the application’s status, rather than error messages, are also useful.

How Log Messages Are Formatted

Every log message has the following four components:

- date and time the message was created
- message type, such as information, warning, or error
- service or application component ID
- message text

When a log message is sent to the text-based destination logs, it is formatted as follows:

```
[Date and time of message] Message type: Service ID: Message text
```

For example, the following messages sent to an ASCII text file illustrate message format:

```
[01/18/00 11:11:12:0] info (1): GMS-017: server shutdown (host
0xc0a801ae, port 10818, group 'iAS') - updated host database
```

```
[01/18/00 11:11:18:2] warning (1): GMS-019: duplicate server (host
0xc0a8017f, port 10818) recognized, please contact iPlanet
Communications for additional licenses
```

Determining the Logging Destination

You can configure the logging service to record server and application messages in any or all of the destinations described in the following table:

Table 4-2 Message Logging Destinations

Log destination	Description	When to use
Process Consoles	iPlanet Application Server process consoles display log messages as they are generated.	This is the default. If logging is enabled and the server is enabled for automatic startup (UNIX) or interaction with the desktop (NT), the consoles open and display the log messages. To disable this feature, deselect the Log to Console checkbox.

Table 4-2 Message Logging Destinations *(Continued)*

Log destination	Description	When to use
Application log	The default application log file. For Windows NT, this can be viewed through the Event Viewer.	This is the default. Warning and information messages are not logged to the application log. The application log provides a more comprehensive record of the server and application error messages. All messages are sorted by their timestamp.
ASCII text file	An ASCII text file, which you must create and specify.	Use when you want a more permanent record of the server and application messages. All messages are sorted by their timestamp.
Database table	A database table which you must create and specify.	This is the most versatile logging destination. Use when you want to sort, group, and create reports of the logged messages.

When you enable logging, the logging service automatically sends messages to the process consoles on Windows and Solaris platforms, as long as those consoles are open and console logging is enabled. On Windows, the logging service also sends messages to the application log. Logging to a process console does not record the messages. You cannot retrieve the messages once they scroll off of the screen.

To enable the logging service and specify the destination of the log messages, perform the following tasks:

1. Click Logging on the iASAT toolbar to open the Logging window.
2. Select the iAS node for which you want to specify log settings.
3. In the right pane of the Logging window, click the Server Event tab.
4. In the Server Event tab, mark the Enable Server Event Log checkbox.
5. In the Log Target box, choose the type of logging to enable, by marking the required checkboxes:
   - Log to Console
   - Log to Windows NT Application Log (Errors Only)
   - Log to File

If you chose to log to a file, that file is created now. See To Rotate Log Files for information about managing log files.

See To Log to a Database in the following section and To Log to a File for more information.

iPlanet Application Server uses a log buffer to store messages before they are written to the application log, an ASCII file, and/or database logs. If a log is updated continuously, the time taken for this effort will negatively affect the performance of the application server. This buffer limits the use of resources by storing messages, till a log is updated. The buffer is written to the destination when either the buffer interval times out or the number of entries in the buffer exceeds the maximum number allowed.

To Log to a Database

If you want to log application server messages to a database, you need to create an event log database table. The following table describes the four field names and lists each field's data type.

NOTE On both Windows and Solaris systems, you can use supplied scripts that automatically set up the eventlog and httplog tables. The scripts are located in the directory `<iASInstallDir>/ias/ias-Samples/dblog/src/schema`, and are named `Log_db2.sql`, `Log_ifmx.sql`, `Log_mssql.sql`, `Log_ora.sql`, and `Log_syb.sql`. Choose the script that is appropriate for the database you're using.

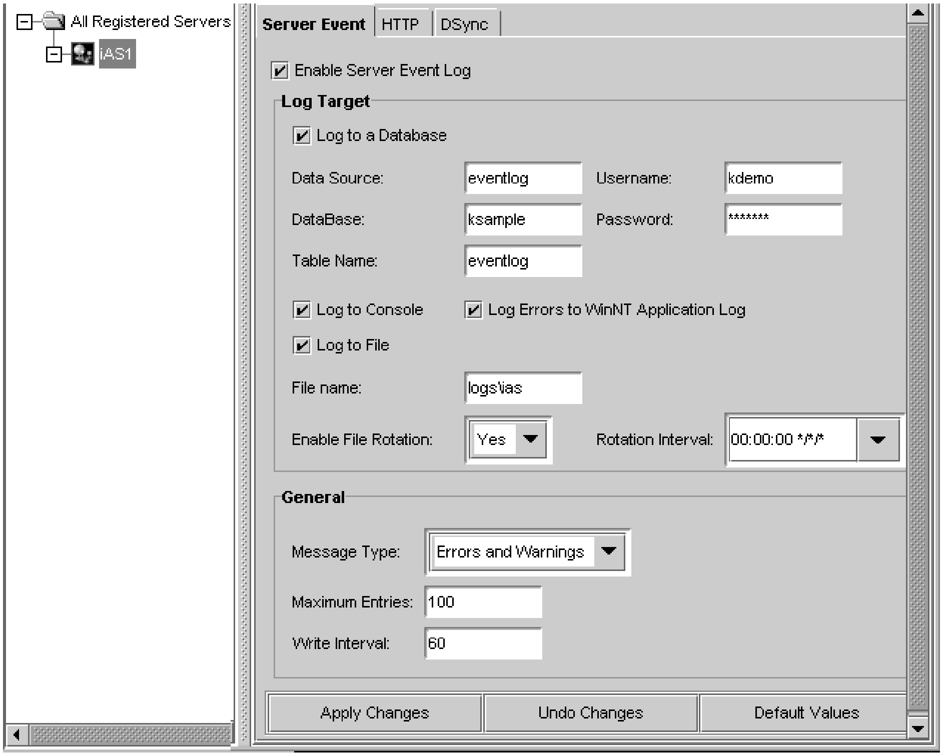
Table 4-3 Logging to a Database Table

Database field name	Description	Data type
`evttime`	Date and time the message was created	Date/Time
`evttype`	Message type, such as information, warning, or error	Number
`evtcategory`	Service or application component ID	Number
`evtstring`	Message text	Text
evtid	Event ID	Number

The logging service maps the message components to the database fields listed in the table. You must use these exact field or column names in your database table.

To log to a database, perform following tasks:

1. Click Logging on the iASAT toolbar to open the Logging window.
2. In the left pane of the Logging window, select the iAS node for which you want to specify log settings.
3. In the right pane of the Logging window, click the Server Event tab.
4. In the Server Event tab, mark the Enable Server Event Log checkbox.
5. In the Log Target box, click the Log to Database checkbox, as shown in the following figure:



☑ Enable Server Event Log

Log Target

☑ Log to a Database

Data Source: eventlog Username: kdemo

DataBase: ksample Password: *******

Table Name: eventlog

☑ Log to Console ☑ Log Errors to WinNT Application Log

☑ Log to File

File name: logs\ias

Enable File Rotation: Yes Rotation Interval: 00:00:00 */*/*

General

Message Type: Errors and Warnings

Maximum Entries: 100

Write Interval: 60

Specify the data source, the database name, the table name, and the user name and password necessary for accessing the database.

6. In the General box, from the Message Type drop-down box, select Errors, Errors and Warnings, or All Messages.
7. Click Apply Changes to save your changes to iASAT.

To Log to a File

iASAT's monitoring service allows you to log information about server activity to a file.

To log information to a file, perform the following tasks:

1. Click Logging on the iASAT toolbar to open the Logging window.
2. In the left pane of the Logging window, click the Server Event tab.
3. In the Server Event tab, mark the Enable Server Event Log checkbox.
4. In the Log Target pane, mark the Log to File checkbox.
5. In the Log to File text field, enter the name of the log file.
6. In the General box, from the Message Type drop-down box, select Errors, Errors and Warnings, or All Messages.
7. Click Apply Changes to save your changes to iASAT.

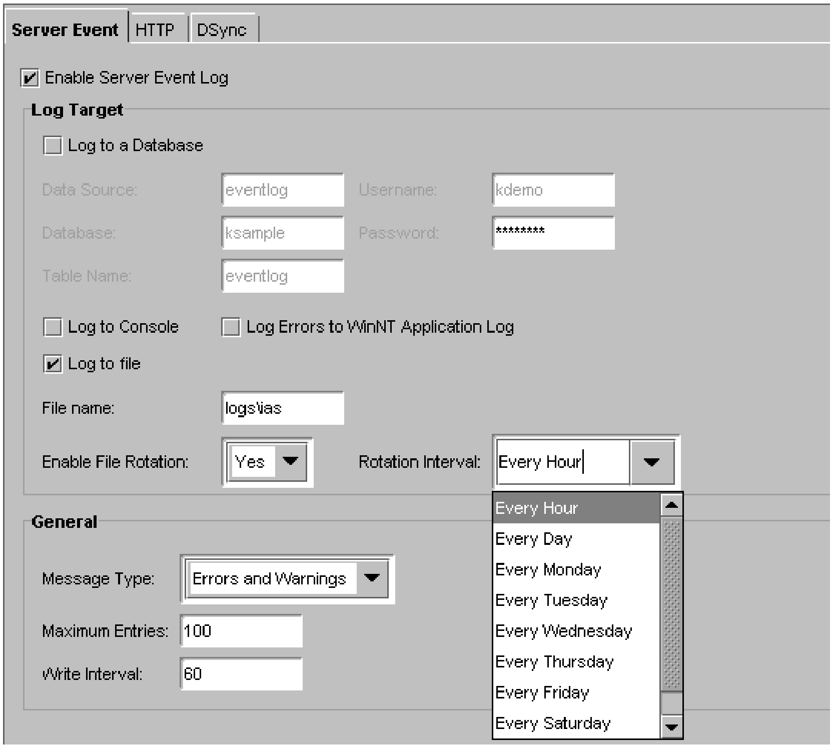
To Rotate Log Files

Log files can be rotated when a set period ends. For example, you can specify that your log files are rotated every 10 days, or every Wednesday, so that only the current logs exist in the log file. Since log files are stamped with the time and date on which they are created, log file rotation helps organize them into manageable units.

If you choose to record server messages in an ASCII file, you can enable log file rotation to regulate the entries periodically.

To configure log file rotation, perform the following tasks:

1. Click Logging on the iASAT toolbar to open the Logging window.
2. Select the iPlanet Application Server node for which you want to specify log settings. The following figure appears:



☑ Enable Server Event Log

Log Target

□ Log to a Database

Data Source:	eventlog	Username:	kdemo
Database:	ksample	Password:	********
Table Name:	eventlog		

□ Log to Console □ Log Errors to WinNT Application Log

☑ Log to file

File name: logs\ias

Enable File Rotation: Yes Rotation Interval: Every Hour

General

Message Type: Errors and Warnings

Maximum Entries: 100

Write Interval: 60

3. In the right pane of the Logging window, click the Server Event tab.
4. In the Server Event tab, select the Enable Server Event Log checkbox.
5. Mark the Log to File checkbox.
6. Choose Yes from the Enable File Rotation drop-down list.
7. Select the interval at which log files are to be rotated, from the Rotation Interval drop-down list. You can also enter a string to indicate when the log file is rotated.

   For instance, the following string indicates logging to a new file begins at 1:00 AM every Monday, as well as on the fifteenth of each month:

   ```
     1:0:0 1/15/*
   ```

   The following string indicates logging to a new file begins at 2:00 AM, 5:00 AM, 6:00 AM, and 7 AM every Friday:

   ```
   2, 5 - 7:0:05/*/*
   ```

8. In the General pane, from the Message Type drop-down box, select Errors, Errors and Warnings, or All Messages.

9. Click Apply Changes to save your changes to iASAT.

About Web Server Requests

You can use the iPlanet Application Server logging service to log web server requests. Web server requests are monitored by the Web Connector Plug-in. The plug-in sends requests to your iPlanet Application Server instance, where they are processed. By logging web server requests, you can track request patterns and other important request information.

This section includes the following topics:

- How Web Requests Are Logged
- To Log Web Server Requests

How Web Requests Are Logged

A web server request is divided into components. These components are standardized HTTP variables used by the web server to manage web requests. iPlanet Application Server includes a subset of these HTTP variables for you to log. You can add variables to the list if you need to log additional information.

NOTE On both Windows and Solaris systems, you can use supplied scripts that automatically set up the HTTP log and event log tables. See To Log to a Database, for more information.

Each HTTP variable must be mapped to a database field name within a table that you create. For instance, to log the length of the content of a web server request, map the `CONTENT_LENGTH` variable to a database field named, for example, `content_length` and defined as a `text` data type. The default HTTP variables used by iPlanet Application Server and their database data types are listed in the following table. Use this table to help you create the database table for logging web requests.

Table 4-4 HTTP Variables and Database Data Types

Default HTTP variables	Default database field name	Data type
Not applicable	logtime	Date/Time
CONTENT_LENGTH	content_length	Number
CONTENT_TYPE	content_type	Text
HTTP_ACCEPT	accept	Text
HTTP_CONNECTION	connection	Text
HTTP_HOST	host	Text
HTTP_REFERER	referer	Text
HTTP_USER_AGENT	user_agent	Text
PATH_INFO	uri	Text
REMOTE_ADDR	ip	Text
REQUEST_METHOD	method	Text
SERVER_PROTOCOL	protocol	Text

You can rename all the database field names except the `logtime` fieldname in the database table. The logging service maps that time that the message was created to the `logtime` database field.

The fields from the database table are automatically mapped to web server variables in the registry.

You must have a web server communication plug-in module such as NSAPI or ISAPI installed and properly configured.

To Log Web Server Requests

Before you can log web server requests, you must create a database table to hold the request messages. For more information about creating this table, see How Web Requests Are Logged.

To log web server requests, perform the following tasks:

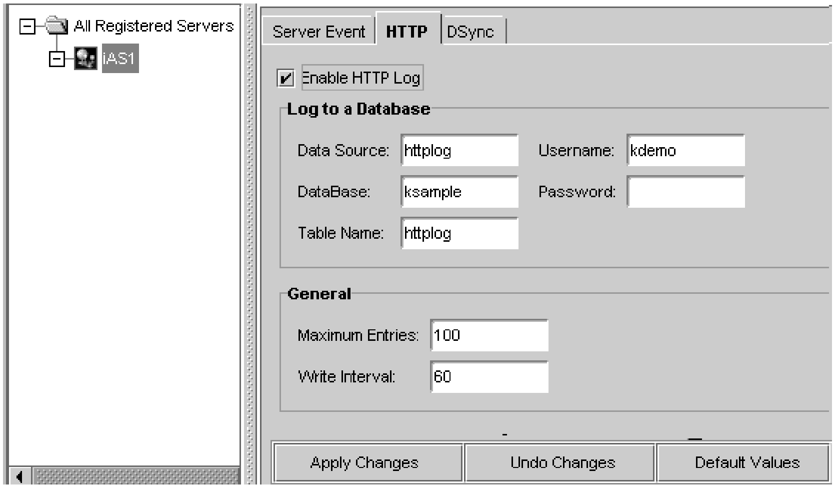
1. Click Logging on the iASAT toolbar to open the Logging window.

2. From the left pane of the Logging window, select the iPlanet Application Server responsible for logging web server requests.

   If you have more than one instance of iPlanet Application Server, you can specify one server to log all web server requests.

3. In the right pane of the logging window, click the HTTP tab.

   The following window appears:



☑ Enable HTTP Log

Log to a Database

Data Source: httplog

Username: kdemo

DataBase: ksample

Password:

Table Name: httplog

General

Maximum Entries: 100

Write Interval: 60

4. Enter `httplog` in the Data Source field.

5. Provide the information you use to connect to the database in the Database field. For example, this would be the Oracle SID for an Oracle database.

6. In the Table Name field, enter `httplog`.

7. Provide the user name and password with which you connect to the database.

8. Enter a number in the Maximum Entries field. This number represents the greatest number of entries that can exist before data is written to the log.

9. Specify the required value in the Write Interval text field.

   This number represents the amount of time that lapses before data is written to the log.

10. Click Apply Settings to save your changes.

About The Administration Server

The administrative services of iPlanet Application Server run in the Administration Server process (KAS) This Administration Server enables remote administration of servers and applications. KAS also supports other services, such as application partitioning, event logging, request monitoring, and dynamic configuration of key server settings. Clients that access administrative services include iASAT, Directory Server and third party SNMP agents.

Some of the events that are monitored and logged by KAS are listed below:

- Server login and logout details.
- Initialization status, startup, shutdown, enabling and disabling of server and engines.
- Issues related to KAS, SNMP, Transaction Manager and engines.
- Issues related to iPlanet Application Server installation.
- Deployment actions and errors during deployment.
- Errors that may occur when retrieving configuration parameters for LDAP, databases, database clients, transactions, resource manager, logging, load balancing, engines and KAS.

About DSync Logging Options

iPlanet Application Server supports Distributed Data Synchronization (DSync) across multiple iPlanet Application Server for partitioned and distributed applications. DSync provides cluster management and data synchronization across iPlanet Application Server processes. iASAT provides for logging of DSync messages.

For more information about distributed data synchronization, see About Distributed Data Synchronization.

This section includes the following topics:

- How DSync Messages Are Logged
- To Log DSync Messages

How DSync Messages Are Logged

DSync provides a component based architecture that allows you to choose which components you want to log. All DSync debug messages appear in KXS, KCS, and KJS log files. DSync debug components are the following:

- ❍ Module: Provides data management and appends other DSync components to the log file. When enabled, the methods executed by DSync are logged.
- ❍ Failover: Provides cluster membership management. When enabled, interactions between servers and how roles change due to failure of servers / engines / network connection are logged.
- ❍ Token: Provides distributed lock management features. When enabled, interactions between servers for read / write tokens associated with DSync nodes are logged.
- ❍ Timeout: Provides life cycle management of DSync nodes per timeout specification. When enabled, nodes that are deleted due to timeout are logged.
- ❍ Messenger: Provides message communication between iPlanet Application Server servers. When enabled, messages that are created, sent, received and processed are logged.

In addition, you can dump cluster and DSync node data into `iasdsync-cluster-XXX.log` and `iasdsync-node-XXX.log` files respectively, where XXX represents the port number of an engine.

Format of the Cluster Dump Files

Each `iasdsync-cluster-XXX.log` file consists of the following sections:

- Cluster
- Message queue

The cluster information reports how an engine views the current Dsync cluster as follows:

```
****************************

*DSync Cluster State

***************************

Host: 0xd00c3643

Port: 10818
```

```
Role: SyncPrimary

Current Engine’s order #:1

SyncPrimary: this engine

Is connect to primary? NO

Changing primary? NO

Max number of SyncBackup#=1

SyncLocal[1]:0xd00c3643:10821

SyncLocal[2]: 0xd00c3643:10822
```

The message queue information displays the list of messages that are in the DSync queues as follows:

```
****************************************

*DSync RecvQueue for GXP_DSYNC protocol

****************************************

Message[1]: GXDSYNC_MSG_RECLAIM_RDTOKEN(/dsync41test/K/5)from
0xd00c3643:10818

...
```

Format of the DSync Node Dump Files

Each `iasdsync-node-XXX.log` consists of the following sections:

- Message queue
- Node Data
- Timeout Manager

The message queue information displays the list of messages that are in the DSync queues as follows:

```
****************************************

*DSync RecvQueue for GXP_DSYNC protocol

****************************************

Message[1]: GXDSYNC_MSG_RECLAIM_RDTOKEN(/dsync41test/K/5)from
0xd00c3643:10818

...
```

The node data section displays the collection of nodes stored in an engine as follows:

```
****************************
*DSync Token State
***************************
[1] ID:/
   Status: without Read or Write Token
   Scope: GLOBAL
[2} ID:/dsync41test
   Status: without Read or Write Token
   Scope: GLOBAL
   Owner Thread: 0xf6f040 (Id=0xf78d50)
   Standard wait queue[1] thread 0xf88670 (Id=0xf883a0)
   Standard wait queue[1] thread 0xf89d60 (Id=0xf89a90)
   Child[0]:B
   Child[1]:A
   Child[2]:D
   Child[3]:C
   Attribute[NextPath]:N
[3]..
```

The timeout manager section displays the set of nodes that are managed by DSync timeout manager in the current engine as follows:

```
****************************************
*Timeout Manager State
****************************************
Entry[0]: ID=/dsync41test/S/4, expired 6 seconds ago
Entry[1]: ID=/dsync41test/U/4, expired 4 seconds ago
Entry[2]: ID=/dsync41test/W/4, expired 3 seconds ago
Entry[3]: ID=/dsync41test/V/4, expired 3 seconds ago
Entry[4]: ID=/dsync41test/X/4, expired 3 seconds ago
Entry[5]: ID=/dsync41test/D/5, expired 2 seconds ago
Entry[6]: ID=/dsync41test/Z/4, expired 2 seconds ago
```

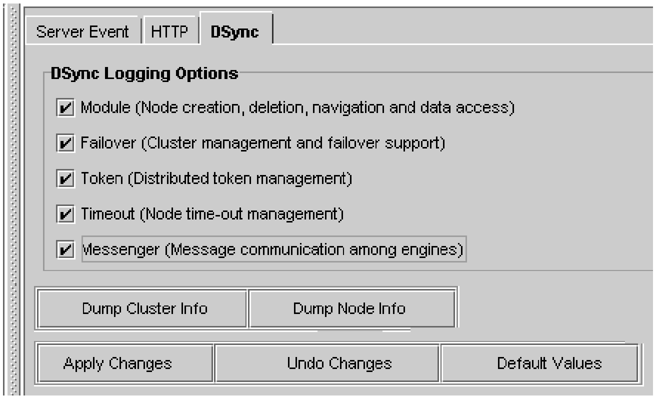
```
Entry[7]: ID=/dsync41test/A/5, expired 1 seconds ago
Entry[8]: ID=/dsync41test/B/5, 0 seconds till expiration
Entry[9]: ID=/dsync41test/C/5, 0 seconds till expiration
Entry[10]: ID=/dsync41test/E/5, 0 seconds till expiration
Entry[11]: ID=/dsync41test/F/5, 1 seconds till expiration
Entry[12]: ID=/dsync41test/H/5, 2 seconds till expiration
```

To Log DSync Messages

To log DSync debug messages, perform the following tasks:

1. Click Logging on the iASAT toolbar to open the Logging window.
2. From the left pane of the Logging window, select the iPlanet Application Server responsible for logging DSync messages.
3. In the right pane of the logging window, click the DSync tab.

   The following window appears:



DSync Logging Options

☑ Module (Node creation, deletion, navigation and data access)

☑ Failover (Cluster management and failover support)

☑ Token (Distributed token management)

☑ Timeout (Node time-out management)

☑ Messenger (Message communication among engines)

4. Specify the DSync components you want to log. See Format of the DSync Node Dump Files for more information on DSync components that you log.

   When specifying DSYNC components, you do not have to shutdown and restart iPlanet Application Server for changes to take affect.

5. (Optional) Click Dump Cluster Info to dump DSync state cluster information to a `iasdsync-cluster-XXX.log` file where XXX is the port number of an engine.

   For information about the format of this log file, see Format of the Cluster Dump Files.

6. (Optional) Click Dump Node Info to dump DSync state node information to a `iasdsync-node-XXX.log` file where XXX is the port number of an engine.

   For information about the format of this log file, see Format of the DSync Node Dump Files.

About Monitoring iPlanet Application Server Log Files

This section describes how application server logs can be viewed, on Windows and Solaris systems. Reviewing log files is often helpful when troubleshooting application and server configuration problems. The following sections describe in detail how you can monitor iPlanet Application Server logs on both Windows and Solaris systems.

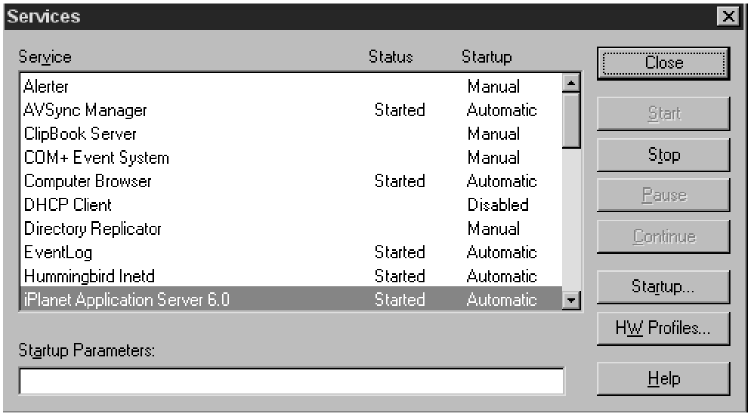
This section includes the following topics:

- To Monitor Application Server Logs on Windows
- To Monitor Application Server Logs On Solaris

To Monitor Application Server Logs on Windows

To enable display of application server logs on NT, perform the following tasks:

1. From the Start menu, select `Settings>Control Panel`
2. Double-click the Services icon. The Services window appears, as shown in the following figure:

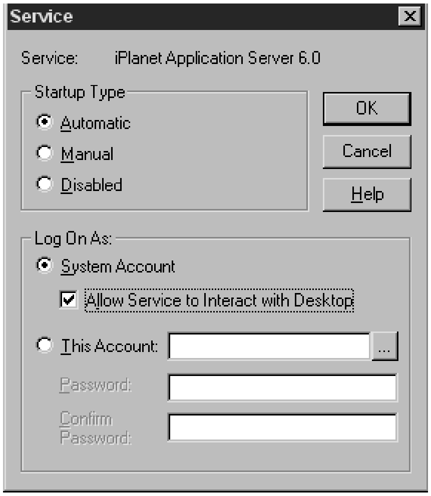


Service	Status	Startup
Alerter		Manual
AVSync Manager	Started	Automatic
ClipBook Server		Manual
COM+ Event System		Manual
Computer Browser	Started	Automatic
DHCP Client		Disabled
Directory Replicator		Manual
EventLog	Started	Automatic
Hummingbird Inetd	Started	Automatic
iPlanet Application Server 6.0	Started	Automatic

Startup Parameters:

3. Select iPlanet Application Server 6.0.

4. Click Startup. The Service window appears, as shown below:



Service: iPlanet Application Server 6.0

Startup Type
● Automatic
○ Manual
○ Disabled

Log On As:
● System Account
☑ Allow Service to Interact with Desktop
○ This Account:
Password:
Confirm Password:

5. Mark the Allow Service to Interact with Desktop checkbox, as shown in the figure.

6. Click OK to close the window.

7. In the Services window, click Stop to stop iPlanet Application Server.

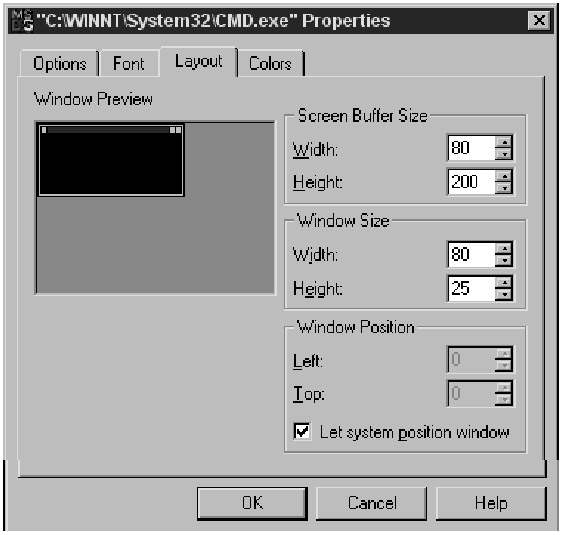
8. Click Start to start iPlanet Application Server again.

When you do this, you will see iPlanet Application Server logs messages displayed in MSDos windows. Each physical process in iPlanet Application Server will be displayed in a separate window. For example, you can see KAS logs, KJS logs and KCS logs in separate windows.

To Enable Vertical Scroll Bars In The Output Windows

The log messages that you view on the MSDos windows will scroll continuously. To You can enable vertical scroll bars in the output windows, and scroll backwards and forwards to view the messages you want. To enable vertical scroll bars in MSDos windows, perform the following tasks:

1. Click the MSDos icon at the upper-left corner of the output window.
2. Choose Properties. The Properties window appears, as shown in the following figure:



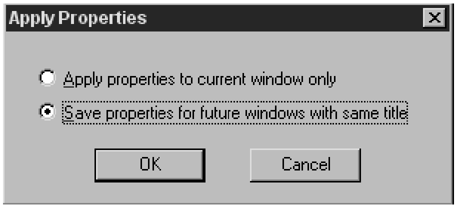
"C:\WINNT\System32\CMD.exe" Properties

Screen Buffer Size
Width: 80
Height: 200

Window Size
Width: 80
Height: 25

Window Position
Left: 0
Top: 0
☑ Let system position window

3. Select the Layout tab.
4. Set the Screen Buffer Size Height to 200 or more.
5. Click OK to register your changes. A confirmation window appears, as shown below:



Apply Properties

○ Apply properties to current window only

● Save properties for future windows with same title

6. To enable scroll bars on all MSDos windows, mark the Save Properties for future windows with same title radio button.

To Monitor Application Server Logs On Solaris

On Solaris, you can view iPlanet Application Server logs by using a command line tool. Using this tool, you can selectively view the messages you want. For example, you can view messages generated during a specific period, or view only the first few messages in a file.

To monitor iPlanet Application Server logs on Solaris, perform the following tasks:

1. Navigate to the following path:

   ```
   <iasinstsall>ias/logs
   ```

2. Type `tail -` and the filename of the log file that you want to view. For example, to view KJS logs, type `tail -kjs.`

To stop the command execution, press `CTRL+C`.

.

NOTE Open a terminal window and type `man tail` for a complete list of all the options you can try with the `tail` command.

Chapter 5

Securing iPlanet Application Server

This chapter describes how to implement iPlanet Application Server security.

The following topics are included in this chapter:

- About Security
- Storing and Managing Users and Groups
- Setting Authorization to Access Application Components
- Enabling Encryption Between Web Server And Application Server
- Using Firewalls for Security
- Configuring Firewalls With iPlanet Application Server

About Security

Implementing application security is a joint effort between the application developers and the server administrator; application developers are responsible for determining the level of security to implement and implementing that level into their applications; the administrator is responsible for managing the users and groups who use the application.

The administrator is also responsible for managing authorization to application components within an application. For Java applications using J2EE standard components, authorization is implemented via roles. Roles are created during deployment time using the iPlanet Application Server Deployment Tool and administered using the iPlanet Application Server Administration Tool (For more information about the Deployment Tool see the online Help system that is provided with the tool.). For C++ applications, authorization is implemented via access control lists that are stored in LDAP and managed using iASAT.

This chapter explains how to set up users and groups and then how they are used to secure applications. It also describes how user entries are stored in iPlanet Directory Server and managed using iPlanet Console and LDIF. The following topics are described in this section:

- Limitations of This Document
- What Is LDAP?
- What Is iPlanet Console?

Limitations of This Document

This chapter provides descriptions of the basic start-up tasks you must perform when setting up Directory Server in association with your instance of iPlanet Application Server, as well as how to use iPlanet Console to manage users and groups. See iPlanet Directory Server and iPlanet Console documentation for detailed instructions and descriptions of these products.

You can find Directory Server documentation installed with your instance of iPlanet Application Server in the following location:

`*iASInstallDir*/manual/en/slapd/`

iPlanet Console documentation is available on iPlanet's web site in the following location:

`http://docs.iplanet.com/docs/manuals/console.html`

What Is LDAP?

Every instance of iPlanet Application Server uses Directory Server to store shared server information, including information about users and groups. Directory Server supports Lightweight Directory Access Protocol (LDAP) versions 2 and 3. LDAP is an open directory access protocol that runs over TCP/IP. It is scalable to a global size and millions of entries. Using Directory Server, you can store all of your enterprise's information in a single, centralized repository of directory information that any application server can access via the network.

iPlanet Directory Server is installed with each instance of iPlanet Application Server.

The types of Directory Servers that can be configured for iPlanet Application Server are as listed below:

Master LDAP Server. The LDAP server that maintains the master data.

Consumer LDAP Server. The LDAP server that contains a copy of the data that is maintained by the master LDAP server. Multiple consumer LDAP servers can be configured for iPlanet Application Server.

Primary LDAP Server. The first LDAP server that is configured to iPlanet Application Server, for configuration information.

Backup LDAP Server. The second LDAP server that is configured to iPlanet Application Server. iPlanet Application Server connects to this server if the primary LDAP server fails. Multiple backup LDAP servers can be configured for iPlanet Application Server.

What Is iPlanet Console?

iPlanet Console is a stand-alone Java application. It finds all resources and applications registered in Directory Server, and displays them in a graphical interface. iPlanet Console functions independently of any server, and you can use it from any computer or workstation connected to your enterprise.

iPlanet Console is installed with each instance of iPlanet Application Server. You use iPlanet Console to manage users and groups for iPlanet Application Server. You can also use iPlanet Console to launch the iPlanet Application Server Administration Tool, but only for local instances of iPlanet Application Server -- that is, instances of iPlanet Application Server installed on the same machine as iPlanet Console. You must launch remote instances of iPlanet Application Server from the command line or from the Windows NT start menu.

Storing and Managing Users and Groups

The information you specify for each user and group you create is stored in the Directory Server. The information held in Directory Server is shared between all application servers when you have multiple servers supporting an application.

This section provides information for the following topics:

- Implementing Security
- Using iPlanet Console to Add Entries to Directory Server
- Using LDIF to Add Entries to Directory Server

- Creating Entries Programmatically

Implementing Security

If access to an application consists of authenticating a user's user name and password, the user name and password must be stored in the Directory Server.

An application starts the user authentication process by calling the application component—usually a servlet—responsible for user authentication. The user's login privileges are then verified against the list of users stored in Directory Server.

The authentication process verifies access to an application based on a user's name and password. To implement authentication, you must create a user profile, which holds the user name and password, for all users of an application. This procedure is described in Using iPlanet Console to Add Entries to Directory Server.

Once a user is successfully authenticated, access to specific application components implementation depends on the type of application: Java application using J2EE standard components or C++ applications.

NOTE There are types of authentication other than verification of user name and password. For example, some applications authenticate a user through a certificate.

Authorization for J2EE Applications

Access to application components responsible for application security is based on declarative role information defined in the deployment descriptor XML file. Security can also be defined programmatically during development by using security APIs such as `isCallerInRole()` provided by J2EE. See the *Developer's Guide (Java)* for more information.

Authorization for C++ Applications

Access to application components responsible for application security is managed declaratively using access control lists provided in the iPlanet Application Server Administration Tool. Security can also be defined programmatically during development by using the LDAP JDK included with each installation of iPlanet Application Server. See the *Developer's Guide (Java)* for more information.

Using iPlanet Console to Add Entries to Directory Server

You can use iPlanet Console to create user entries and group entries. A user entry contains information about an individual person or object in the directory. A group consists of all users who share a common attribute. For example, all users in a particular department might belong to the same group.

What Is a Distinguished Name (DN)?

Each of the users and groups in your enterprise is represented in Directory Server by a distinguished name (DN). A DN is a text string that contains identifying attributes. You use DNs whenever you make changes in the directory's users and groups database. For example, you need to specify DN information each time you create or modify directory entries, set up access controls, and set up user accounts for applications such as mail or publishing. The users and groups interface of iPlanet Console helps you create or modify DNs.

For example, this might be a typical DN for an employee of iPlanet Communications Corporation:

```
uid=doe,e=doe@iplanet.com,cn=John Doe,o=Netscape Communications
Corp.,c=US
```

The abbreviations before each equal sign in this example have the following meanings:

- `uid`: user ID
- `e`: email address
- `cn`: the user's common name
- `o`: organization
- `c`: country

DNs may include a variety of name-value pairs. They are used to identify both certificate subjects and entries in directories that support LDAP.

Creating User Entries Using iPlanet Console

User security is best suited for applications that have a small number of known users. You must create a user profile for each user who accesses the application.

You must be a Directory Server administrator or a user with the necessary permissions to create a user.

To create a new user entry in the directory using iPlanet Console, perform the following steps:

1. On Windows Systems.

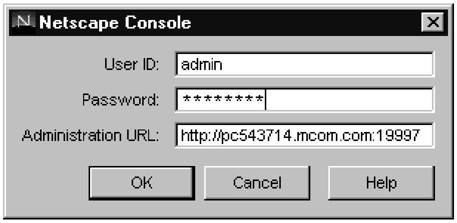
   From the Windows Start menu, click `Programs>iPlanet Server Products> Select iPlanet Console 4.0`.

   On Solaris Systems.

   Navigate to the <`iASInstall`> path, and type the following command:

   `./startconsole`.

   The iPlanet Console login dialog box appears:



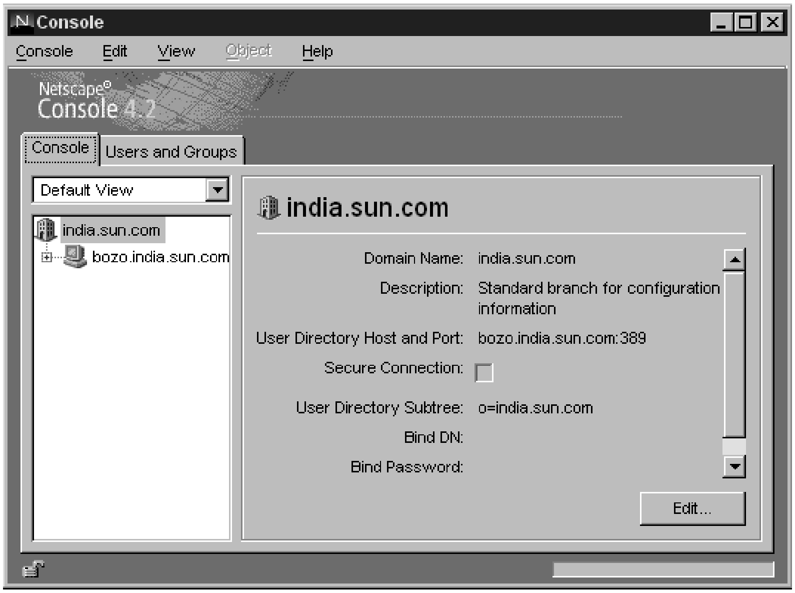
User ID: admin

Password: ********

Administration URL: http://pc543714.mcom.com:19997

2. In the User ID text field, provide the user name that was specified during iPlanet Application Server installation.

3. In the Password field, type in the password that you provided for the iPlanet Console Administration Server, during iPlanet Application Server installation. iPlanet Console's main window appears, as shown in the following figure:



india.sun.com

Domain Name: india.sun.com

Description: Standard branch for configuration information

User Directory Host and Port: bozo.india.sun.com:389

Secure Connection: □

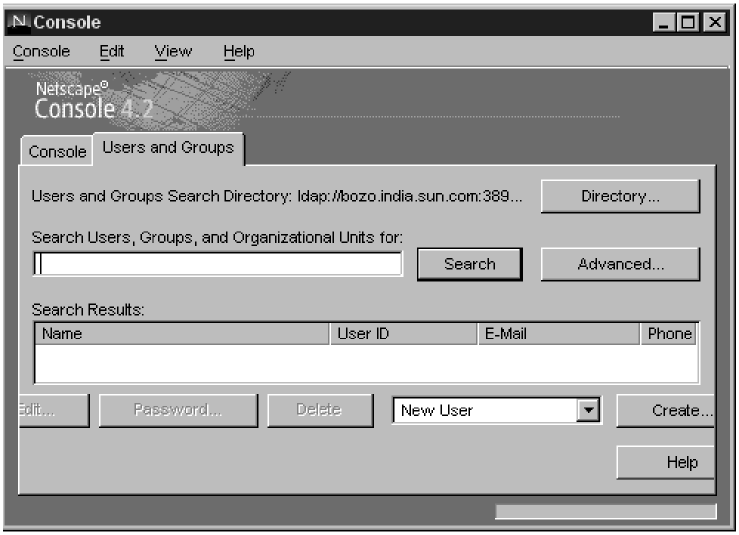
User Directory Subtree: o=india.sun.com

Bind DN:

Bind Password:

4. Click the Users and Groups tab.

   The following window appears:



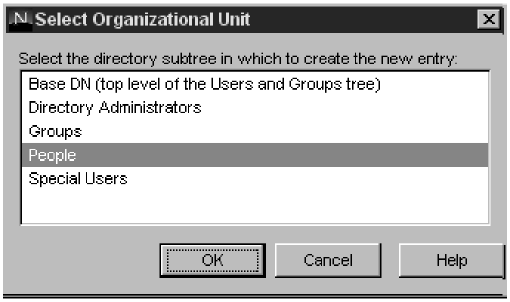
Users and Groups Search Directory: ldap://bozo.india.sun.com:389...

Search Users, Groups, and Organizational Units for:

Search Results:

5. Choose New User from the drop-down list in the lower-right corner of the window, and then click Create.

The Select Organizational Unit dialog box appears, as shown in the following figure:

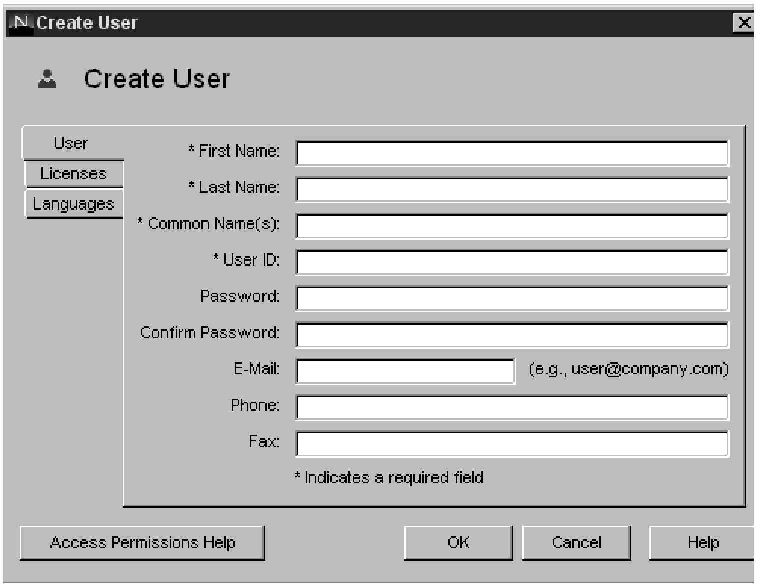


Select the directory subtree in which to create the new entry:

Base DN (top level of the Users and Groups tree)
Directory Administrators
Groups
People
Special Users

6. In the Select Organizational Unit window, click the directory subtree (ou) to which the user will belong, then click OK.

   The Create User window appears, as shown in the following figure:



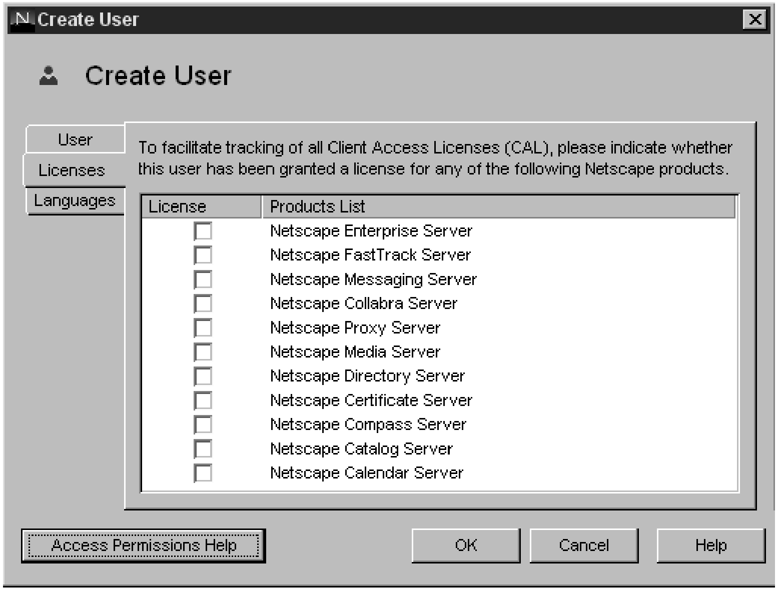
7. In the Create User window, enter user information.

NOTE

- The full name of the user is equivalent to the common name (cn) in the directory and is automatically generated based on the First Name and Last Name entered above. You can edit this name later, if required.
- A user ID is automatically generated from the first and last names you enter. You can replace this user ID with one of your choosing. The user ID must be unique from all other user IDs in the directory.

8. Click the Licenses tab when you finish entering user information in the Create User window.

The following window appears:



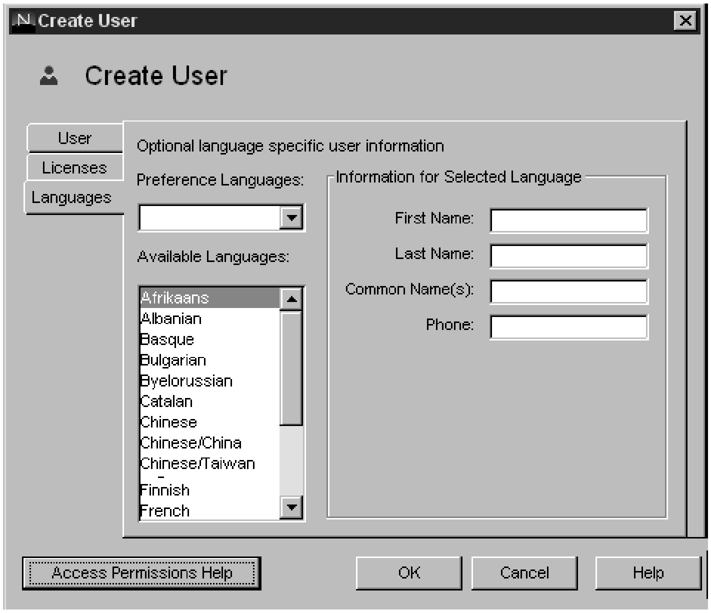
To facilitate tracking of all Client Access Licenses (CAL), please indicate whether this user has been granted a license for any of the following Netscape products.

License	Products List
□	Netscape Enterprise Server
□	Netscape FastTrack Server
□	Netscape Messaging Server
□	Netscape Collabra Server
□	Netscape Proxy Server
□	Netscape Media Server
□	Netscape Directory Server
□	Netscape Certificate Server
□	Netscape Compass Server
□	Netscape Catalog Server
□	Netscape Calendar Server

9. Select the servers the user you are creating is licensed to use.

10. (Optional) Click the Languages tab.

The following window appears:



- Use the Preference Languages drop-down list to select the user's preferred language. Select a language to see the Pronunciation field when appropriate.
- Enter language-related information.
- Click OK.

Creating Group Entries Using iPlanet Console

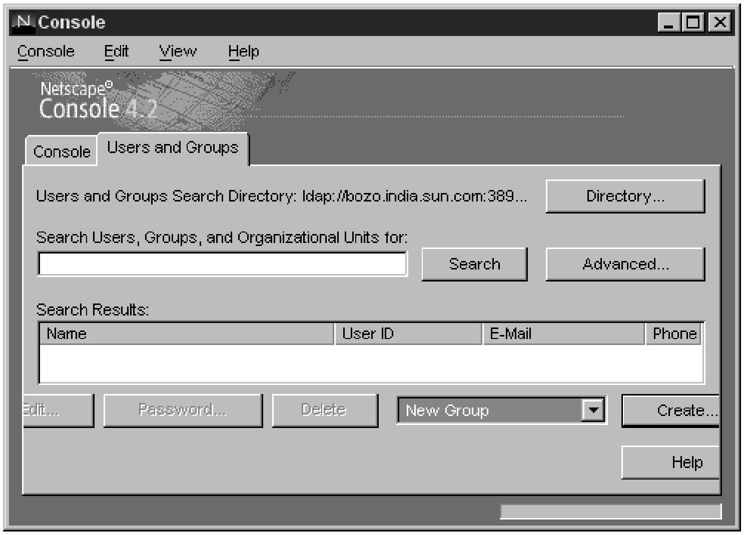
A group consists of all users who share a common attribute. For example, all users with DNs containing the attribute `ou=Sales` belong to the Sales group. Once you create a new group, you add users, or members, to it. You can use three types of groups in your directory: static, dynamic, and certificate groups.

Creating a Static Group

Create a static group by specifying the same group attribute in the DNs of any number of users. A static group doesn't change unless you add a user to it or delete a user from it. For example, a number of users can have the attribute `department=marketing` in their DN. These users are not members of the Marketing group, until you explicitly add each user to the that group.

To create a static group in the directory, perform the following steps:

1. In iPlanet Console, click the Users and Groups tab to display the following window:



Users and Groups Search Directory: ldap://bozo.india.sun.com:389...

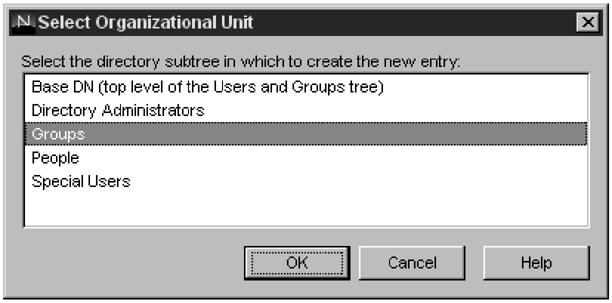
Search Users, Groups, and Organizational Units for:

Search Results:

Name	User ID	E-Mail	Phone

2. Choose New Group from the drop-down list in the lower-right corner of the window and then click Create.

The following dialog box appears:

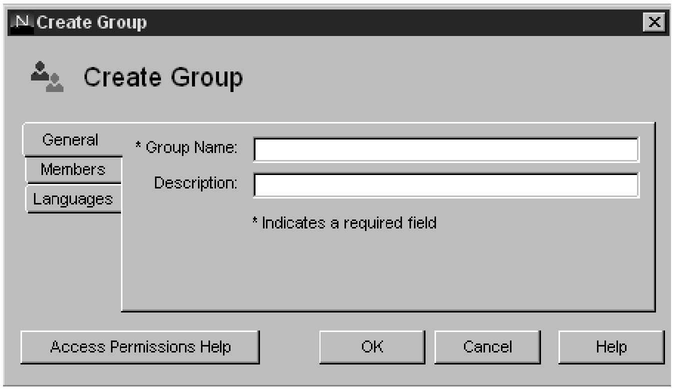


Select the directory subtree in which to create the new entry:

- Base DN (top level of the Users and Groups tree)
- Directory Administrators
- Groups
- People
- Special Users

3. In the Select Organizational Unit window, select the directory subtree (ou) to which the group will belong, then click OK.

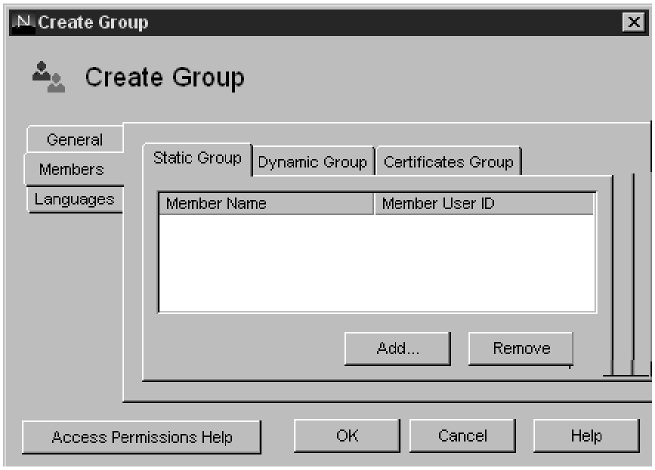
The Create Group window appears:



* Indicates a required field

4. In the Create Group window, enter group information, then click the Members tab.

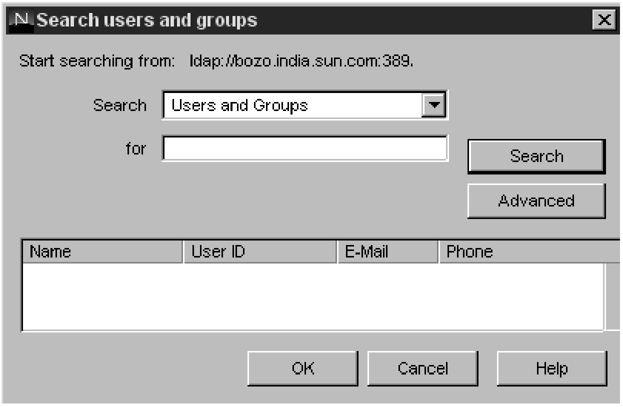
   The following window appears:



5. If you only want to create the group now and plan to add group members later, click OK and skip the rest of this procedure.

   To immediately add members to the group, continue to the next step.

6. Click Add. The Search users and groups dialog box appears:


Search users and groups

Start searching from: ldap://bozo.india.sun.com:389.

Search: Users and Groups

for:

Locate a user you want to add to the Members User ID list and click Ok. Repeat this step until all the users you want to add to the group are displayed in the Member User ID list.

Modifying Database Entries Using iPlanet Console

Before you can modify user or group data, you must first use the Users and Groups Search function to locate the user or group entry in the user directory. You can then select operations from the menu toolbar to change the entry. The operations you perform apply to all users in the Search list.

See iPlanet Console documentation for more information.

Using LDIF to Add Entries to Directory Server

Directory Server uses the LDAP Data Interchange Format (LDIF) to describe a directory and directory entries in text format. LDIF is commonly used to initially build a directory database or to add large numbers of entries to the directory all at once. You can also add or edit entries using the `ldapmodify` command along with the appropriate LDIF update statements.

To add entries to the database using LDIF, first define the entries in an LDIF file, then import the LDIF file from Directory Server.

Formatting LDIF Entries

LDIF consists of one or more directory entries separated by a blank line. Each LDIF entry consists of an optional entry ID, a required distinguished name, one or more object classes, and multiple attribute definitions.

The basic form of a directory entry represented in LDIF is:

```
dn: distinguished name

objectClass: object class

objectClass: object class

...

attribute type[;subtype]:attribute value

attribute type[;subtype]:attribute value

...
```

You must supply the DN and at least one object class definition. In addition, you must include any attributes required by the object classes that you define for the entry. All other attributes and object classes are optional. You can specify object classes and attributes in any order. The space after the colon is also optional. For information on standard object classes and attributes, refer to the iPlanet Directory Server documentation at:
`http://docs.iplanet.com/docs/manuals/directory.html`

Modifying Database Entries Using ldapmodify

You use the `ldapmodify` command-line utility to modify entries in an existing Directory Server database. `ldapmodify` opens a connection to the specified server using the distinguished name and password you supply, and modifies the entries based on LDIF update statements contained in a specified file. Because `ldapmodify` uses LDIF update statements, `ldapmodify` can do everything that `ldapdelete` can do. Most of Directory Server's command-line utilities are stored in a single location. You can find them in the following directory:

```
<iASInstallDir>/bin/slapd/server
```

The command line tools `ldapdelete`, `ldapmodify`, and `ldapsearch`—are stored in the following directory:

```
<iASInstallDir>/shared/bin
```

The following is an example of the command used to add a user to an LDIF file:

```
ldapmodify -h myserverhost -p 389 -D “Directory Manager” -w admin -a
-f MyUsersFile
```

Creating Entries Programmatically

You can also create entries programmatically within an application using the LDAP JDK included with each installation of iPlanet Application Server. See the *Developer's Guide* (Java) for more information.

Setting Authorization to Access Application Components

Authorization to access application components depends upon the type of application:

- For Java Applications (using J2EE standard components), authorization is set via roles. See Setting Role-Based Authorization (for J2EE Applications).
- For C++ Applications, authorization is set by permissions in access control lists. See Setting Access Control List Authorization (for C++ Applications).

This section includes the following topics:

- Setting Role-Based Authorization (for J2EE Applications)
- Setting Access Control List Authorization (for C++ Applications)

Setting Role-Based Authorization (for J2EE Applications)

Roles for an application component are set globally for all application components within a module. From the Administration Tool, you can add a role to an application module and set the users and groups who belong to a role. Access is granted to any application component within a module if the requestor is a member of a pre-defined role.

If a user is not a member of a role, the application can direct the user to re-login, prompt the user to exit the application, or direct the user to a different part of the application.

Managing Roles for EJBs and Servlets

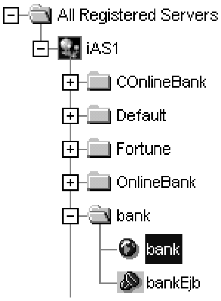
You need to use iPlanet Application Server Administration Tool (iASAT) to manage the roles of deployed applications. When you manage roles, you can specify groups to which users belong and add only groups to the role rather than adding individual users as members to the role. This is useful if you are using individual user-based security; you save the administration maintenance of updating users in the role when users change.

For example, if you have created users for an web bank application and a user closes all accounts, you need to remove that user only from the appropriate group or groups, as opposed to removing the user from the groups and any roles.

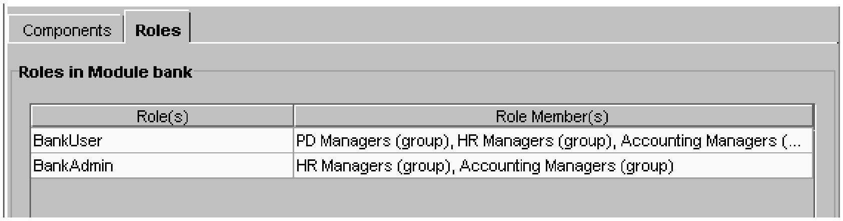
NOTE Roles for servlets and EJBs are created in the deployment descriptor XML files before deployment. See the online help that is provided with the Deployment Tool for more information.

To manage a role, perform the following steps:

1. On the iASAT toolbar, click Application to open the Application window.
2. In the left pane of the Application window, expand the iPlanet Application Server instance where the application is deployed.
3. Open the application folder and highlight a servlet or EJB icon, as shown in the following figure:



4. In the right pane of the Application window, click the roles tab to view the roles and role members that have been defined for this EJB/servlet.

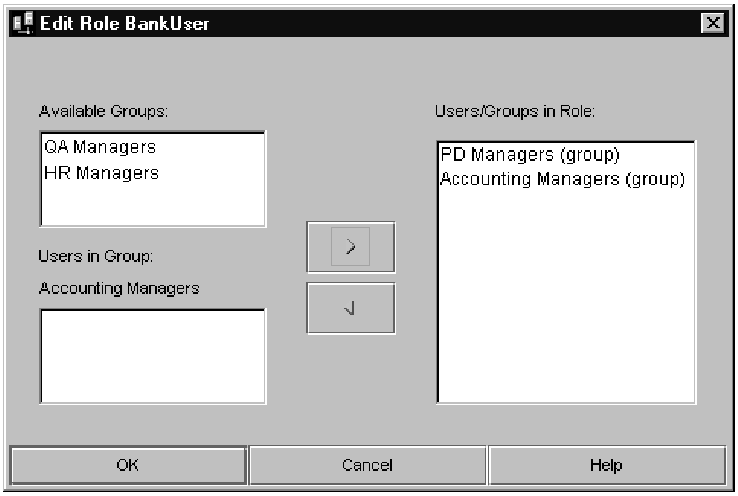


Roles in Module bank

Role(s)	Role Member(s)
BankUser	PD Managers (group), HR Managers (group), Accounting Managers (...
BankAdmin	HR Managers (group), Accounting Managers (group)

5. Highlight the role that you want to manage and click Edit Role, at the bottom of the window.

   The Edit Role dialog box opens showing you all the users and groups that are currently members of this role.



Edit Role BankUser

Available Groups:
QA Managers
HR Managers

Users in Group:
Accounting Managers

Users/Groups in Role:
PD Managers (group)
Accounting Managers (group)

6. To add a group and a user to a role, complete the following:

   a. To add a group to a role, in the Available Groups box, highlight one or more groups and click the right-arrow button.

NOTE When you select multiple groups from the Available Groups box, the users in the Available Users box are not displayed.

b. To add a user to a role, first highlight a group to which the user currently belongs from the Available Groups list and then highlight the user(s) in the Users in Group box. Finally click the right-arrow button to add the user to the role.

7. To remove a group or user from a role, highlight the user(s) and or group(s) in the Users / Groups in Role box and click the left-arrow.

Setting Access Control List Authorization (for C++ Applications)

Access control lists (ACLs) allow you to set permissions for users and groups. A permission relates to an action the user is allowed to perform, such as read or write.

iPlanet Application Server comes with default permissions, but you can also create your own application-specific permissions and ACLs. The information in an ACL is used by the application to verify the permissions of the current user or group for an action the user attempts.

If a user does not have a certain permission, the application can direct the user to re-login, prompt him to exit the application, or direct him to a different part of the application.

Creating an Access Control List

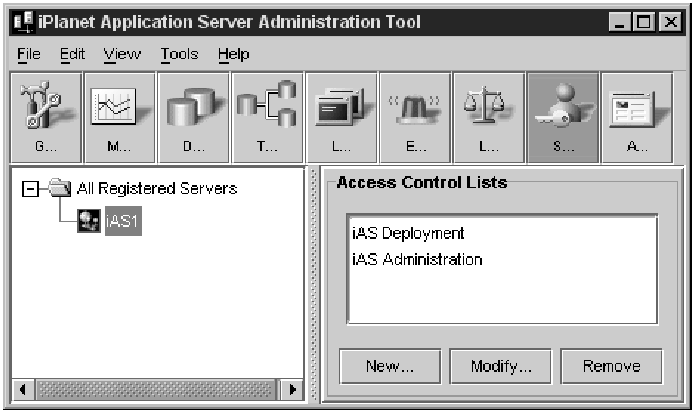
You need to use iASAT to create and manage access control lists (ACLs). When creating an ACL, you can create groups to which users belong and add only groups to the ACL rather than adding individual users as members to the ACL. This is useful if you are using individual user-based security; you save the administration maintenance of updating users in the ACL when users change.

For example, if you have created users for an intranet application and a user leaves the company, you need to remove that user only from the appropriate group or groups, as opposed to removing the user from the groups and any ACLs.

To create an access control list, perform the following steps:

1. On the iASAT toolbar, click Security to open the Security window.

   The following window appears:

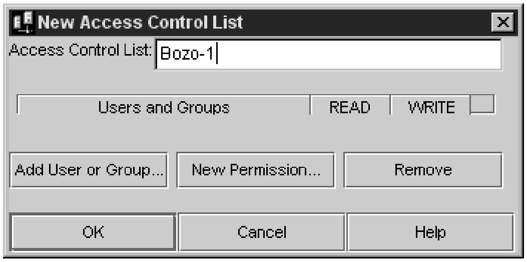


Access Control Lists:

iAS Deployment

iAS Administration

2. Click New. The New Access Control List dialog box appears, as shown in the following figure:



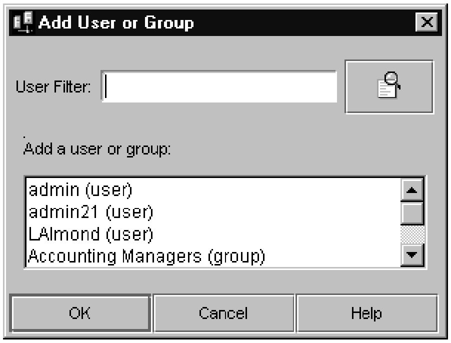
Access Control List: Bozo-1

3. In the Access Control List text field, enter a name for the ACL.

   The name can be any word or words you choose to distinguish one ACL from another.

4. To add a user or group to the ACL, click Add User or Group.

   The Add User or Group dialog box appears.



admin (user)
admin21 (user)
LAlmond (user)
Accounting Managers (group)

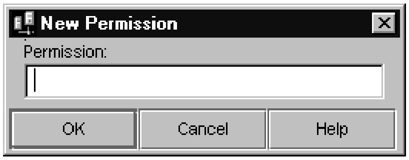
5. Select the users and/or groups you want to add to the ACL.

   You can filter the list of users that appears in the result set by entering a string in the User Filter text box. For instance, to show only user IDs that begin with "F," enter `F*` in the User Filter text box, then click the icon next to the User Filter text box. The user IDs matching your filter criteria appear in the list box below. The User Filter applies only to users, not to groups.

6. Click OK.

7. To add a new permission to the ACL, click New Permission.

   The New Permission dialog box appears.



8. Provide a new permission action word.

   A permission defines the level of access a user or group has to a particular application or part of an application.

9. Click OK.

10. To set the appropriate permissions for the groups in the ACL, check each permission for that group.

Modifying an Access Control List

You can modify the following ACL properties:

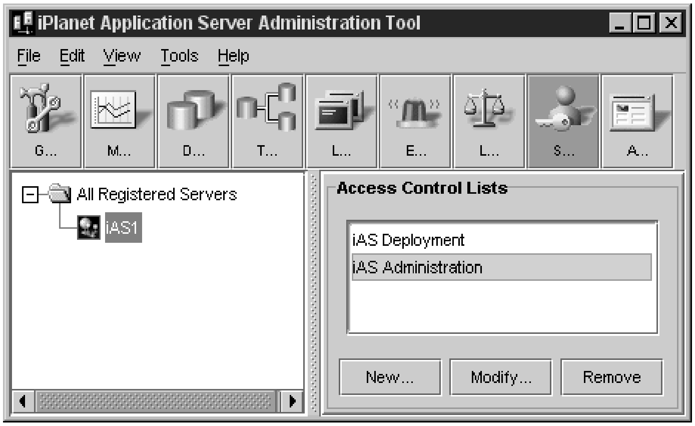
- add groups
- create new permissions
- edit permissions

You can also remove groups from the system.

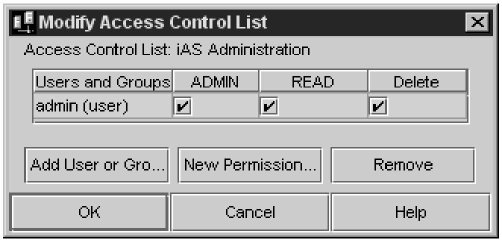
To modify an access control list, perform the following steps:

1. On the iASAT toolbar, click Security to open the Security window.

   The following window appears:



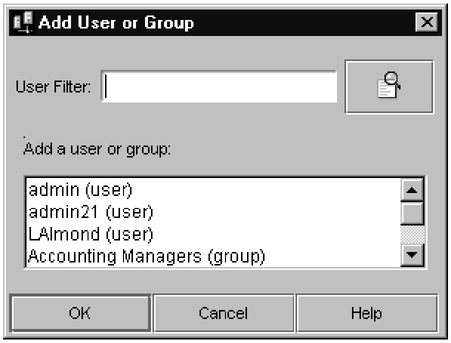
2. Highlight the Access Control List that you want to modify.
3. Click Modify. The Modify Access Control List dialog box appears.



Access Control List: iAS Administration

Users and Groups	ADMIN	READ	Delete
admin (user)	☑	☑	☑

4. To add a new user or group, click Add User or Group. The Add User or Group dialog box appears.



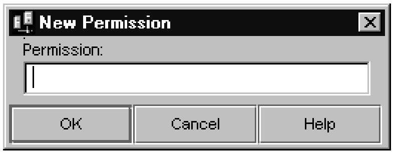
5. Select the group or groups you want to add to the ACL.

   You can filter the list of users that appear in the list by entering a string in the User Filter text box. For instance, to show only user IDs that begin with "F," enter `F*` in the User Filter text box, then click the icon next to the User Filter text box. The user IDs matching your filter criteria appear in the list box below. The User Filter applies only to users, not to groups.

6. Click OK.

7. In the Modify Access Control List window, click New Permission to create a new permission.

   The New Permission dialog box appears.



8. Provide a new permission such as Read, Write, in the Permission text field and click OK.

9. In the Modify Access Control List window, you can edit the permissions for a group, by selecting or deselecting the checkboxes next to the permissions.

10. To remove a group, select that group and click Remove in the Modify Access Control List window.

Enabling Encryption Between Web Server And Application Server

You can selectively encrypt the traffic between Web Server(s) and Application Server(s). The Executive Server (KXS) manages the traffic between the servers. You may need to enable encryption for components that require high security, such as servlets that gather credit card information, login servlets, etc. Components are encrypted using 128 bit keys and RSA BSafe 3.0 library, which ensures safe transfer of information.

This section also describes how you can verify the encryption details in KXS logs and how to create a cryptext key for encryption.

The following topics are included in this section:

- To Enable Encryption Between Web Server And Application Server
- To Verify The Encryption Log Messages
- To Enable Encryption For Each J2EE Component

To Enable Encryption Between Web Server And Application Server

To enable encryption of traffic between Web Server(s) and Application Server(s), perform the following steps:

1. Open iPlanet Registry Editor.

   (See About iPlanet Registry Editor)

2. Open the following key:

   `SOFWARE/iPlanet/Application Server/6.0/CCSO/Security/`

3. You will see a value that looks like this; `EnableEncryption=0.` By default, encryption is disabled. To enable encryption, modify this value to read `EnableEncryption=D` (D stands for Domestic 128 bit data type string). Encryption is now enabled.

To Verify The Encryption Log Messages

To verify the encryption log messages in the KXS log file, perform the following steps:

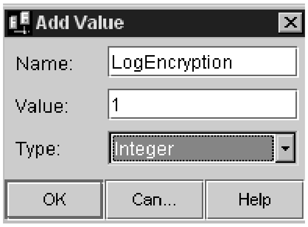
1. Open iPlanet Registry Editor.

   (See About iPlanet Registry Editor)

2. Open the following key:

   `SOFWARE/iPlanet/Application Server/6.0/CCSO/Security/`

3. From the Edit menu, choose `Add Value`. The Add Value dialog box appears, as shown in the following figure:



Add Value

Name: LogEncryption

Value: 1

Type: Integer

4. In the Name field, type `LogEncryption`.
5. In the Value field, type `1`.
6. From the Type drop-down list, select Integer.

When you create this key, encryption log messages will be logged in the KXS logs.

To Enable Encryption For Each J2EE Component

To enable encryption for individual J2EE components such as JSPs and Servlets (encryption cannot be enabled for EJBs) perform the following steps:

1. Open the `ias-web.xml` file of the application for whose component you want to enable encryption.
2. Search for and locate the servlet or JSP for which you want to enable encryption.

3. Locate the `<encrypt>false</encrypt>` tag for the component. Change *false* to *true*.

4. Register the application using `iasdeploy`.

To verify that the encryption is enabled and working, open `KXS` logs and search for messages that are similar to the following:

```
[11/Jan/2001 19:58:43:0] info: CRPT-001: Encrypting 2309 bytes,
keysize = 128 bits

[11/Jan/2001 19:58:43:5] info: NSAPICLI-012: plugin reqstart,
tickct: 1903570535

[11/Jan/2001 19:58/:43:5] info: NSAPICLI-009: plugin reqexit:
Os+.12995s. (198114 0537)

[11/Jan/2001 19:58:52:2] info: CRYPT-004: Decrypting 1897 bytes,
keysize = 128 bits
```

Note that if you enable encryption for a component, encryption will take place even if it has been disabled in iPlanet Registry.

Using Firewalls for Security

iPlanet Application Servers are typically deployed together with web servers and client databases applications. In this scenario, inter-process communication is of a very high level and this makes and data integrity and security issues very important. To ensure smooth and secure communication between application servers, web servers and the client applications that run on application servers, security components of the system are frequently separated by firewalls.

This section describes how you can effectively plan and deploy iPlanet Application Servers in fire-wall protected environments. The following topics are described in this section:

- Basics of Network Security
- iPlanet Application Server Architecture
- Inter-Process Communication Protocol
- Encrypting Data Channels

NOTE All the addresses and port numbers listed in this document are default values. These may be changed during install, or later by modifying values stored in the appropriate registries. If these are changed, then the appropriate modifications will need to be applied to firewall configuration parameters.

Basics of Network Security

Private networks, such a company's Local Area Network (LAN) are not directly connected to public networks such as the Internet, but pass through a firewall which is intended to control access. Access to public networks is controlled using approved protocols on the approved port numbers of specific machines.

Types of Firewalls

There are many different types of firewalls, each with its own set of advantages and disadvantages. Firewalls do not necessarily consist of simply one type of machine, but typically involve other machines that may be present. These other systems may configured to provide basic traffic routing, but may or may not be configured to provide firewall functionality. The three main types of firewalls are described in the following sections:

Routers

These are fast, and in the simpler versions, relatively cheap. At their most basic, they sit between two networks, and relay connections between them. Configuration parameters can be specified to allow or disallow connections to or from individual IP Addresses or ranges of IP Addresses.

More complex versions of routers will inspect data to determine the security protocol being used, and the port number to which the security protocol is addressed. Control can be exercised on connections using specific protocols and directed to specific ports (filters).

Routers can be machines built specifically for the task, or can be implemented using a computer (typically running UNIX operating system) with multiple network interfaces. The advantage of using a general purpose machine as a dedicated router is that the router can also run proxy servers.

Proxy Servers

Proxy Servers are normally used in conjunction with some form of router to ensure that data access can be only through the proxy server. A proxy server usually sits on a machine that either straddles the firewall, or more frequently, sits behind a primary router on a small, dedicated network often known as a De-Militarized Zone (DMZ), with another router between the proxy server and the private network. The proxy system is generally a very basic, stripped down system which has few or no normal user applications, just programs which act as proxy servers for various protocols.

A proxy server is designed to act as a server to some client inside the private network. For example an HTTP proxy server will behave as an HTTP server as far as a web browser on the private network is concerned. The browser will connect to the HTTP proxy server and send a request for a document. The HTTP proxy server will then in turn act as a browser and will connect to the server from which the document has been requested. The remote HTTP server is not aware of the proxy server. When the HTTP proxy server gets the requested document, it scans the document for viruses before passing it on to the users' browser on the private network. The actions that the HTTP proxy server performs before surrendering the document to the browser can be specified in the HTTP proxy server. This enables additional security and also ensures the quality and wholesomeness of information.

Stateful Inspection Firewall

An increasingly popular form of firewall is a machine, which doesn't simply route information, but also inspects the contents of information packages. Using the information obtained, higher level protocols can be determined. This can extend right up to the level of application protocols. Once the communication protocols have been established the system uses state tables to determine which operations to allow until the protocol enters a new state.

For example, let us assume that an incoming telnet connection will be a TCP/IP (Transmission Control Protocol/Internet Protocol) connection directed to port number 23 on the destination machine. The telnet server will respond with the number of a port, to which the telnet client should connect. The statefull inspection firewall system will recognize the telnet opening state, and examine the returned data for the port number to which the client should connect. When the host (the machine that initiated the connection) directs the connection to the destination port, the statefull inspection firewall system will allow the connection. Other random connections to non-standard ports will normally be denied.

Statefull inspection firewalls are fast because not all data has to be examined. They can be used for general purpose security, unlike proxy servers, which need to be specifically created to handle a given protocol. Creating a proxy for a new protocol can also take a significant amount of time. With a statefull inspection firewall system, a new state table can be created quickly and easily, often by simply monitoring the new protocol to determine appropriate state transitions and actions.

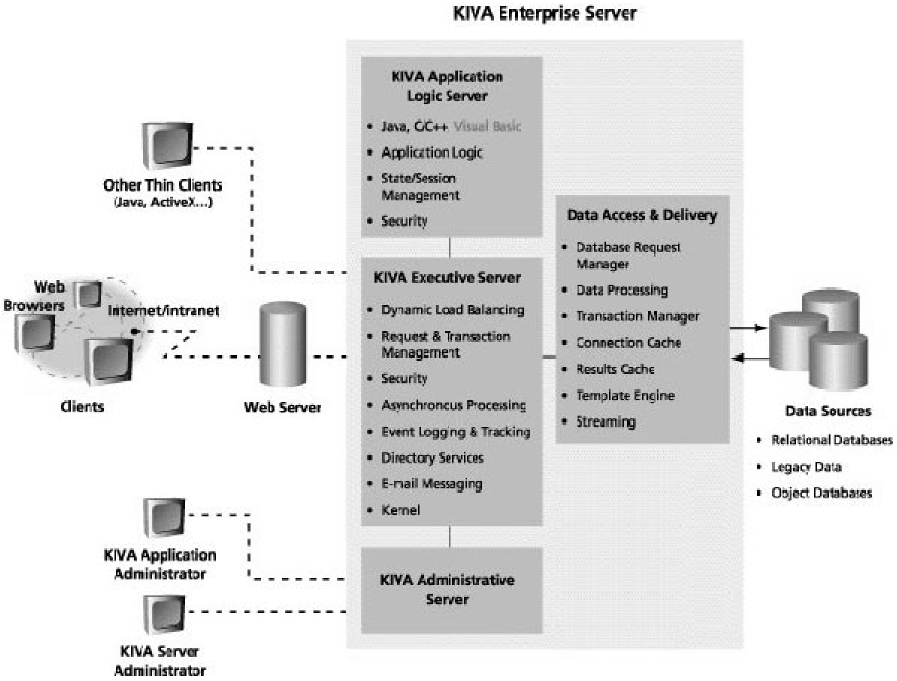
The ensuing sections describe the architecture of iPlanet Application Server and the how communication takes place within iPlanet Application Server.

iPlanet Application Server Architecture

iPlanet Application Server is composed of several modules. The basic component is the Executive Server (KXS) which creates the components of the application, manages per session data, load monitoring and load-balancing with other instances of iPlanet Application Server.

The following diagram illustrates the basic iPlanet Application Server architecture.

NOTE You will see the name Kiva mentioned several times in the following diagram. This is because the original name of iPlanet Application Server was “Kiva”. Therefore, some iPlanet Application Server components are preceded with the name Kiva, for example KJS (Kiva Java Server) KCS (Kiva C++ Server) and KXS (Kiva Executive Server).



The application code runs in multi-threaded processes created by KXS. There are two types of processes; the C++ Server (KCS) and the Java Server (KJS).

The system is managed through the Administration Server (KAS). The web server may be installed on the same machine as iPlanet Application Server, but will typically be installed on a separate machine.

The communication between the web server and iPlanet Application Server is through a plug in, residing in the web server which communicates directly with iPlanet Application Server. If multiple instances of iPlanet Application Server have been installed, the plug in communicates with the iPlanet Application Server selected by the load balancing system. In cases where the web server in use does not have a plug in available, the communication module may be called as a CGI (Common Gateway Interface) application, which establishes connection to iPlanet Application Server. The use of the CGI model is less efficient, and normally used only as a last resort.

The second method of communication is using OCL (Object Constraint Language), which uses CORBA to locate the required services and communicate with them through iPlanet Application Server. This is recommended only for the intranet because of the lack of standardized security for IIOP (Internet Inter-Object Protocol) connections.

The next section describes how a firewall is structured and how communication takes place between servers, applications and security protocols.

Inter-Process Communication Protocol

Inter-process communication between servers, applications and security protocol occurs in a number of ways. The communication links and protocols involved are as given below:

Ephemeral Ports

Ephemeral ports are the anonymous ports created in response to a request from a client to a server. In UDP (User Datagram Protocol), these will be the port numbers used for reply datagrams. For TCP/IP, these will be the port numbers for the connection that a server creates in response to a request from a client.

Ephemeral ports can be any ports not currently in use, and not in the reserved range (`0–1023`). It is common practice to allocate these ports from the range `(1024-5000)`. However, some systems, notably Solaris, now allocate ephemeral port numbers that belong to a different range. Typically any port number greater than 32768 can be allocated as an emphemeral port, going by the recognition that for large systems with many connections the limited range of port numbers may not be sufficient.

It is important to verify the values used and to ensure that the appropriate range or ranges are enabled for ephemeral ports on any firewalls where ephemeral port usage exists. Typically this will be `> 32768` for modern Solaris systems and `1023` to `5000` for Windows systems.

Transmission Control Protocol/Internet Protocol (TCP/IP)

There are multiple TCP/IP connections carrying different types of traffic as listed in the following table. The addresses and ports indicated in the table are default values:

Connection	Traffic	Port/Addr	Purpose

Web Connector Plug-in to KXS	Request data	10818	User I/O
KXS to KCS/KJS	Internal data	10819 (*)	Send requests
CGI to KXS	Request data	See Text	User I/O
KCS/KJS to KXS	Internal data	10818	Results/Admin
Admin UI to KAS	Internal data	10817	Admin.

NOTE (*) Begins at this number and increases by for each KCS/KJS.

Most of these communication channels are used for more than one type of data transfer. Data transfers are multiplexed over the same TCP/IP connection.

When a web server other than the web servers that support a plug in is used, communication with KXS will be through small CGI applications.

The port used for communication between CGI to KXS is not fixed, but is allocated as:

```
10819 + (number of KCS/JKS engines)
```

Therefore, if 4 KJS engines and one KCS engine are installed, the CGI port will be configured at port number `10824`, that is default CGI port number `10819+3`, during installation of iPlanet Application Server.

TCP/IP communication for web servers is allocated on port `80` or `443` depending on the use of HTTP or HTTPS (Secure HTTP).

IP Multicast

In the case of multiple iPlanet Application Server instances, iPlanet Application Server uses 2 IP multicast channels, which handle the following communication tasks:

- IP multicast channel 1 communicates load balancing information between KXS processes.
- IP multicast channel 2 communicates administration information and commands between KAS processes.

It should be noted that traffic on the KXS channel can be heavy, and increases with system load. Default addresses for KXS channels are given in the following table:

MCast	Address	Port
KXS <-> KXS	228.8.18.71	9607
KAS <-> KAS	228.8.18.71	9608

NOTE The multiplexing of internal server data over a single TCP/IP channel is sometimes referred to as the Kiva Communications Protocol (KCP). This protocol is meant only for internal data transfer.

User Datagram Protocol (UDP)

The Web Connector Plug-in uses UDP. It uses the Ping protocol to verify that the KXS process with which the Web Connector Plug-in is currently associated is still alive. A thread running within the KXS responds to the ping. Response to a ping indicates that the KXS is running. Lack of response indicates that failover should be initiated. to switch to an alternate KXS. Default UDP ports are given in the following table:

UDP Packet	Port
Ping request	9610
Ping response	Ephemeral

Where UDP is not permissible, the ping may be deactivated by modifying the Web Connector Plug-in's iPlanet Registry. To disable response to a ping request, perform the following tasks:

1. Start iPlanet Registry

   (See About iPlanet Registry Editor)

2. Open the following key:

   ```
   SOFTWARE/iPlanet/Application Server/6.0/CCS0/CONN/
   ```

3. Modify the `DisableEcho` key. Set the value to 1.

TIP To modify a value in iPlanet Registry, perform the following steps:

a. Double-click the key name or select the key and choose Modify Value from the Edit menu, to bring up the Modify Value dialog box.

b. Enter the new value in the dialog box.

c. Click OK to register the change in iPlanet Registry.

When you turn UDP pinging off, the currently selected iPlanet Application Server may fail and this could cause a delay, until the TCP/IP connection times out. This may take up to several minutes, depending upon TCP/IP configuration parameters.

Note that if the network disables UDP, the ping must be disabled in the Web Connector Plug-in registry for the Web Connector Plug-in to work. UDP ping is used only by the web server plug in and not by the CGI interface.

The following table provides a summary of the protocols, addresses and purpose of iPlanet Application Server components:

What	Protocol	Port/Addr	Purpose
Web Connector Plug-in	TCP/IP	10818	Web Connector Plug-in to KXS
Web Connector Plug-in/CGI	TCP/IP	> 1024	Return connection
Web Connector Plug-in	UDP/IP	9610	Keepalive ping
Web Connector Plug-in	UDP/IP	Ephemeral	Ping return
KXS to KCS/KJS	TCP/IP	10819	KXS to KCS/KJS
CGI to KXS	TCP/IP	Last KCS/KJS + 1	CGI apps. to KXS
KCS/KJS to KXS	TCP/IP	10818	KCS/KJS to KXS
KXS	IP Multicast	228.8.28.71:9607	KXS to KXS
KAS	IP Multicast	228.8.18.71:9608	KAS to KAS
Admin UI to KAS	TCP/IP	10817	Admin.

Encrypting Data Channels

You can encrypt communication that occurs between the Web Connector Plug-in and KXS. When you do this, the same ports and addresses continue to be used, but data transfers are encrypted. This is recommended on an absolute need-basis, because of the overhead associated with data encryption using secure cyphers. See Enabling Encryption Between Web Server And Application Server.

Configuring Firewalls With iPlanet Application Server

This section examines some common firewall configurations and the parameters to be configured for correct functioning. These examples illustrate where firewalls can reasonably be placed within a iPlanet Application Server implementation.

Although it is possible to place a firewall between iPlanet Application Server instances (an iPlanet Application Server instance being a KXS/KJS/KCS process group) this is not recommended. If such a firewall is implemented it is essential that it implements IGMP (Internet Group Management Protocol) to allow IP across the firewall.

Although firewalls are not recommended between iPlanet Application Server instances, the implementation of a fast, dedicated network for IP Multicast traffic between iPlanet Application Server machines may be advisable in some circumstances.

The iPlanet Application Server system is designed for networks running at LAN speeds, so distributing iPlanet Application Server instances across a WAN (Wide Area Network) may lead to performance problems.

Single Firewall

The simplest and most common firewall configuration is with a firewall between the web server and the Internet as shown in the following illustration:

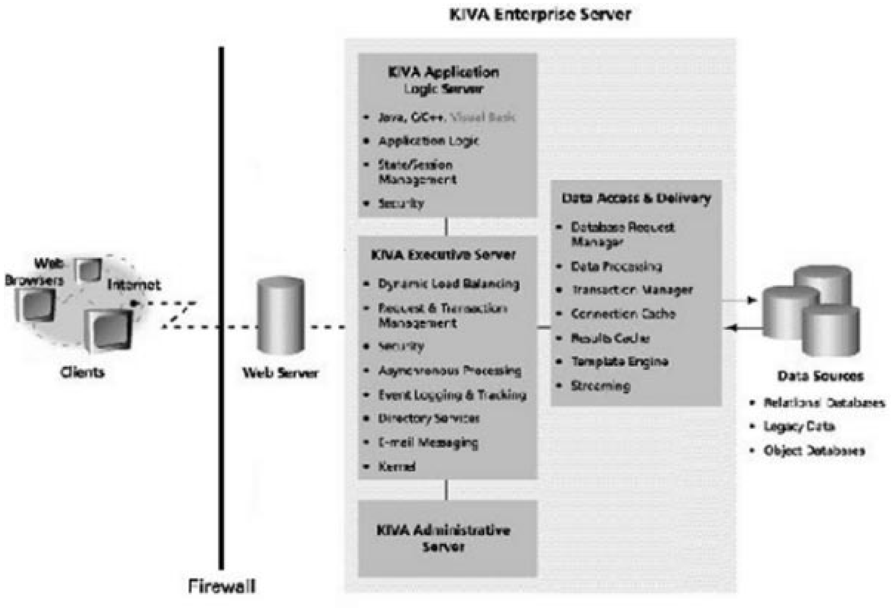


Figure 5-1 Single Firewall Between the Web Server and the Internet - Example 1

In this case, firewall configuration is straightforward. The firewall needs to be configured to allow HTTP connections to port 80 and/or HTTPS 443 as appropriate. Return data is on a port above 1024, so these ports need to be opened for outgoing (reply) connections.

The advantage of this configuration is simplicity. The biggest disadvantage is that there is a single line of defense. Once the firewall is breached the only defense is the security of each of the individual machines within the private network.

The table below summarizes the protocols and ports which need to be configured to the firewall to permit correct functioning.

Protocol	Direction	Port	Reason
TCP/IP	Incoming	80	HTTP requests
TCP/IP	Incoming	443	HTTPS requests
TCP/IP	Reply (out)	>1024	HTTP(S) response

Sometimes, it may be required to run the web server outside the firewall, as shown in the following illustration:

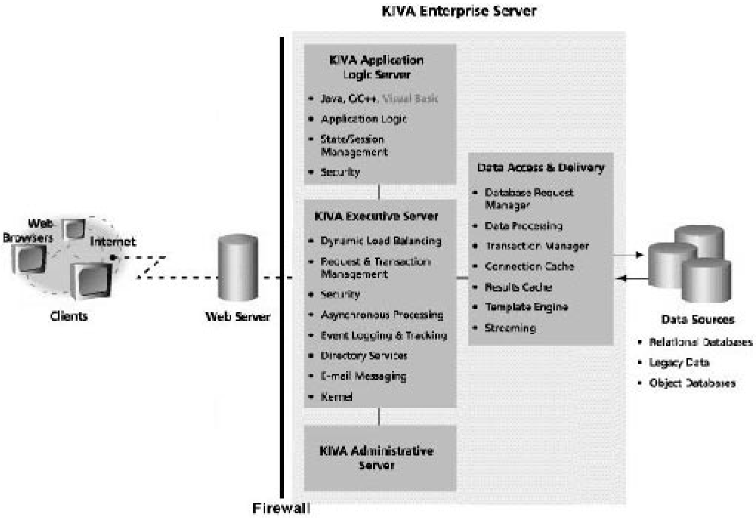


Figure 5-2 Single Firewall Between the Web Server and the Internet -Example 2

This configuration exposes the web server machine to the Internet, which is not recommended. As in the previous example, there is a single line of defense. The external web server will typically have some level of trust within the protected network, and so would be a prime target for harmful intent.

This firewall configuration needs to allow the Web Connector Plug-in TCP/IP data connection and UDP ping connection, as described in the following table:

Protocol	Direction	Port	Reason
TCP/IP	Incoming	10818	Web to KXS
TCP/IP	Incoming	See note below	CGI to KXS
TCP/IP	Reply (out)	> 1024	KXS to Web
UDP	Incoming	9610	Web to KXS
UDP	Reply (out)	Ephemeral	KXS to web

NOTE If a Web Connector Plug-in is not available for the web server in use, CGI applications will be used to communicate with the KXS. The CGI port allocated by default at installation time is: `10819 + (number of KCS/KJS instances)`

Double Firewall - the DMZ Configuration

This configuration is becoming more popular as many enterprises open up limited access to their private networks to partners and customers via DMZs:

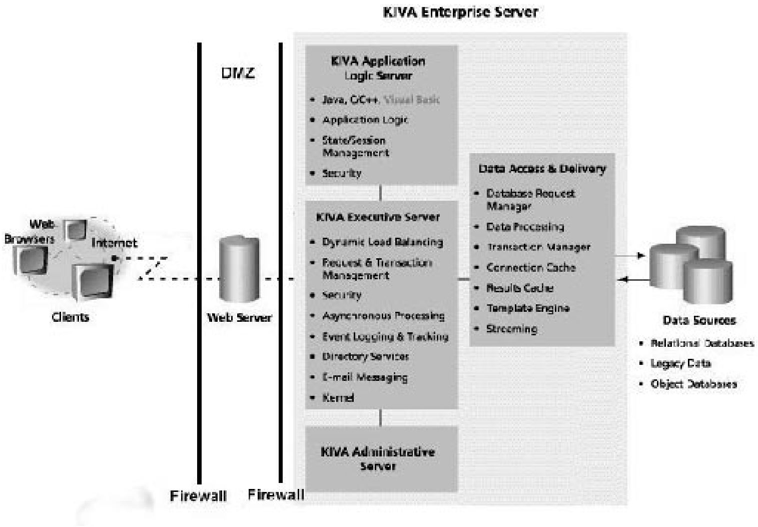


Figure 5-3 Double Firewall - the DMZ Configuration

The security provided by this configuration is much better than the single firewall examples above. The double layer of protection combined with active monitoring of activity at each firewall and within the DMZ will detect most attempts to penetrate the internal network.

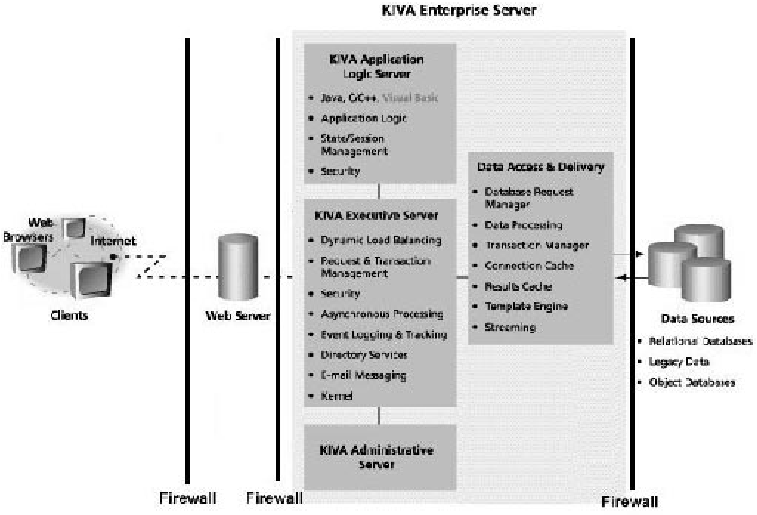
In this case the outer firewall will need to be configured as in the above example to allow HTTP and HTTPS transactions. The inner firewall will need to be configured to allow the web server plug in to communicate with the iPlanet Application Server server(s) behind the firewall.

The default port used by the web plug in to communicate with the KXS instances is 10818. Similarly, the UDP port used for the failover keepalive pings is 9610 by default, changing this requires appropriate modification the firewall configuration. The following table lists the default port numbers for both KXS ports and the UDP ports:

Protocol	Direction	Port	Reason
TCP/IP	Incoming	10818	Web to KXS
TCP/IP	Reply (out)	> 1024	KXS to Web
UDP	Incoming	9610	Failover ping
UDP	Outgoing	Ephemeral	Failover ping reply

Triple Firewall - DMZ With Database Protection

In some corporate settings, databases reside on their own networks, protected by firewalls as shown in this diagram.



This configuration provides maximum security for what may be the most important corporate asset; the data contained within the corporate database. The firewall between the LAN and the database systems provides protection against internal as well as external threats.

The connections to the databases use standard access mechanisms such as ODBC (Open DataBase Connectivity), JDBC (Java DataBase Connectivity) and database vendor supplied connector libraries. The connections to the database are not different from any other application, and so the firewall configuration for the database protection layer will conform to the standard settings required for access to the specific database in use.

NOTES *FTP Access to web server*

When an application is deployed, it will typically have a few HTML pages that must be installed in the web server document hierarchy. The installation of these pages is done manually, usually transmitting the pages through FTP (File Transfer Protocol) to the web server(s). When configuring any firewall between iPlanet Application Server and its web server(s) care must be taken to allow FTP (or some other means of copying files) from iPlanet Application Server to the web server. This needs to be permitted only during deployment of an application and can be disabled at other times.

Remote Administration

In order to run iPlanet Application Server Administration Tool remotely, the network connection needs to allow TCP/IP on port 10817 for the connection between iASAT and KAS.

Chapter 6

Enabling High Availability of Server Resources

This chapter describes how to enable high availability of iPlanet Application Server resources, to ensure effective system performance.

Increasing iPlanet Application Server resources such as the number of threads, processes, and restart attempts can increase the performance of the applications running on the server, and reduce the likelihood of application downtime.

You need to consider the resources of the iPlanet Application Server instance before you plan on increasing server resources. For instance, if the iPlanet Application Server instance is working at full capacity, adding to the load can negatively affect the performance of an application. Likewise, assigning additional threads to a process removes available threads from the system-wide thread pool, limiting the system's ability to process other thread-utilizing requests, such as database access.

Directory Server stores the bulk of information on which iPlanet Application Server's usage depends. It is important, therefore, that this information is not lost, if Directory Server fails due to some reason. To ensure high availability of a huge volume of information and configuration data, it is necessary to configure a backup Directory Server to take over when the primary Directory Server fails.

This chapter deals mainly with how you can effectively configure iPlanet Application Server's resources and ensure availability of data.

This chapter includes the following topics:

- About Adding and Tuning Server Processes
- Adjusting the Number of Threads for Processing Requests
- Specifying the Number of Requests for the Executive Process
- Setting Options of the Administration Server

- Implementing a Multi-Process, Single-Threaded Environment
- Configuring Directory Server Failover

About Adding and Tuning Server Processes

You can add a Java Server (KJS) or C++ Server (KCS) process to increase high availability. When you add one or two additional processes, an application is more likely to respond to requests. For instance, if one process fails, the second or third process can take its place, decreasing the period for which an application is unavailable. This is particularly useful for applications that have known problems which can cause a process to fail.

In addition, you can add a Bridge process to enable direct communication to application components hosted by a KJS process on iPlanet Application Server, using RMI/IIOP. When a request originates from a Rich Client, it is sent to iPlanet Application Server through a Bridge process. This allows Rich Clients to communicate directly to application components on iPlanet Application Server. For more information on adding a bridge process, see To Add a CXS Process.

This section describes how to accomplish the following tasks:

- To Add and Tune Java and C++ Processes
- To Add a CXS Process

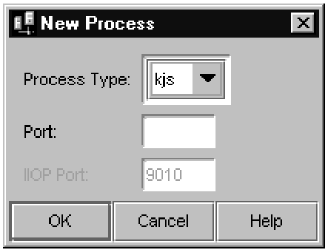
To Add and Tune Java and C++ Processes

You can add additional KJS processes for Java applications and KCS processes for C++ applications. It may not be necessary to add more than two processes for either type of application. If an application cannot run on one or two processes, you could check for errors in the code, which can be addressed by the application developer.

To add a Java or C++ process, perform the following tasks:

1. On the iASAT toolbar, click General to open the General window.
2. In the left pane of the General window, select the iPlanet Application Server instance where you want to add the KJS process.
3. From the File menu, click `New>Process`.

   The New Process dialog box appears.



Process Type: kjs

Port:

IIOP Port: 9010

4. From the Process Type drop-down list, choose KJS or KCS.
5. In the Port Number text box, specify an unused port number where the additional process will run.
6. Click OK to add the new process.
7. If this process is to be used in a single-threaded environment, perform the following tasks:
   a. Select the required process in the left pane of the General window.
   b. In the right pane of the window, set the Minimum and Maximum Threads to 1.
8. Click Apply Changes to save your changes.

To Add a CXS Process

You must add a CXS (Bridge) process if Rich Clients are to communicate directly with EJBs hosted on a KJS process via the Internet Inter-ORB Protocol (IIOP). Typically, requests are made through a web path where they originate at a Web Browser and are then processed by JSPs and servlets, which in turn access EJBs. This web path uses the HTTP protocol. In the case of Rich Clients, requests are made through a Java program directly to EJBs using the CORBA Executive Server (CXS), a Java engine within iPlanet Application Server, which acts as a bridge between Rich Clients and EJBs. For more information about Rich Clients see the *Developer's Guide (Java).*

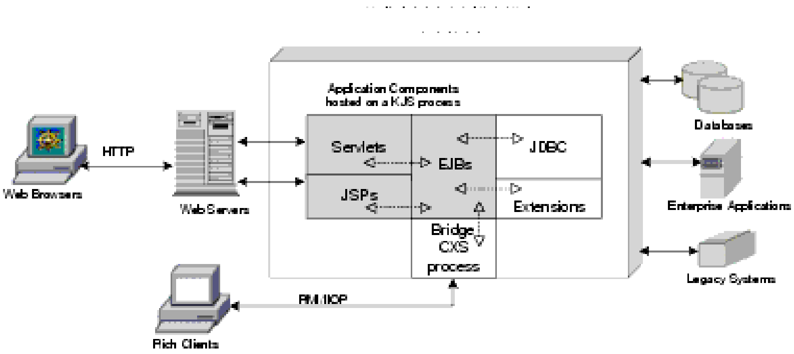
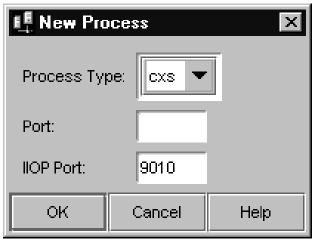


Figure 6-1 iPlanet Application Server Communication Architecture

To add a CXS (Bridge) process, perform the following tasks:

1. On the iASAT toolbar, click General to open the General window.
2. In the left pane of the General window, select the iPlanet Application Server instance where you want to add the CXS process.
3. From the File menu, click `New>Process`.

   The New Process dialog box appears.



New Process

Process Type: cxs

Port:

IIOP Port: 9010

4. From the Process drop-down box, choose CXS.
5. In the Port text box, specify an unused port number where the additional process will run. This is an internal iPlanet Application Server engine port.
6. In the IIOP Port text box, specify a port number to be used by the Rich Client to talk to CXS. This is the port in which CXS listens for the Rich Client.
7. Click OK to add the new process.

8. To use this process in a single-threaded environment, perform the following tasks:

   a. Select the required process in the left pane of the General window.

   b. In the right pane of the window, set the Minimum and Maximum Threads to 1.

9. Click Apply Changes to save your changes.

Adjusting the Number of Threads for Processing Requests

Request threads handle user requests for application components. When iPlanet Application Server receives a request, it assigns the request to a free thread. The thread manages the system needs of the request. For example, if the request needs to use a system resource that is currently busy, the thread waits until that resource is free before allowing the request to use that resource.

You can specify the minimum and maximum number of threads that are reserved for requests from applications. The thread pool is dynamically adjusted between these two values. The minimum thread value you specify holds at least that many threads in reserve for application requests. That number is increased up to the maximum thread value you that you specify.

Increasing the number of threads available to a process allows the process to respond to more application requests simultaneously. You can add and adjust threads for each process, or you can define the number of threads for all processes under a server, at the server level.

By default, each process uses the threads assigned to iPlanet Application Server. For example, if iPlanet Application Server uses a minimum of 8 threads and a maximum of 64 threads, each individual process uses a minimum of 8 threads and a maximum of 64 threads.

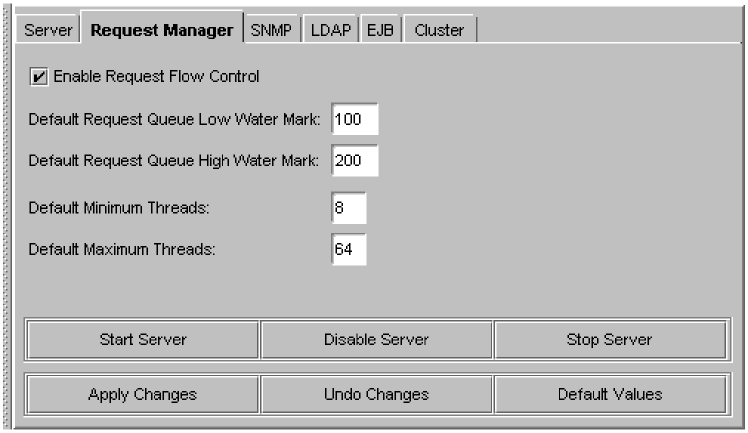
This section describes the following topics:

- To Adjust Threads at Server Level
- To Adjust Threads at Process Level

To Adjust Threads at Server Level

To adjust the number of request threads for all (KJS / KCS / KXS and IIOP) processes for a server, perform the following tasks:

1. On the iASAT toolbar, click General to open the General window.
2. From the left pane of the General window, select the server for which you want to adjust the number of threads.
3. Click the Request Manager tab in the right pane of the General window.
4. In the Default Minimum Threads text box, specify the minimum number of threads available for each process of the selected iPlanet Application Server.



☑ Enable Request Flow Control

Default Request Queue Low Water Mark: 100

Default Request Queue High Water Mark: 200

Default Minimum Threads: 8

Default Maximum Threads: 64

5. In the Default Maximum Threads text box, enter the maximum number of threads available for each process on the selected iPlanet Application Server.
6. Click Apply Changes to save your changes.

To Adjust Threads at Process Level

You can also specify different thread settings for each process under a server. Note that the numbers you specify for a process will be accepted as default for that process. The process level setting overrides the server level setting.

To adjust the number of threads available for a process under a server, perform the following tasks:

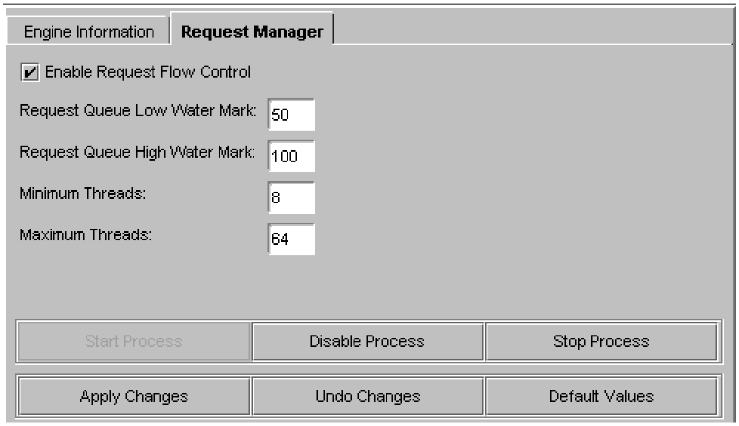
1. On the iASAT toolbar, click General to open the General window.

2. In the left pane of the General window, select the server whose process you want to edit. Expand the server to view its processes. Note that you can expand a server's hierarchical tree only when it is running.

3. Select the process whose number of threads you want to adjust.

4. Click the Request Manager tab, from the right pane of the General window.

5. Specify the minimum number of threads available for the selected process, in the Minimum Threads text box.



☑ Enable Request Flow Control

Request Queue Low Water Mark: 50

Request Queue High Water Mark: 100

Minimum Threads: 8

Maximum Threads: 64

6. Specify the maximum number of threads available for that process, in the Maximum Threads text box.

7. Click Apply Changes to save your changes.

NOTE Settings that are specified at process level override those set at the server level.

Specifying the Number of Requests for the Executive Process

The Web Connector Plug-in routes users requests aimed at iPlanet Application Server applications, to the Executive process (KXS). These requests are logged to the request queue in the Executive process.

You can perform the following tasks:

- Control the maximum number of threads the Web Connector Plug-in will use to process requests. This prevents the request queue from receiving more requests than it can process.
- Set the maximum number of requests that are logged to the request queue to control the flow of requests. The maximum number is called the “high watermark”.
- Set the number of requests in the queue in which logging will resume. This number is called the “low watermark”.

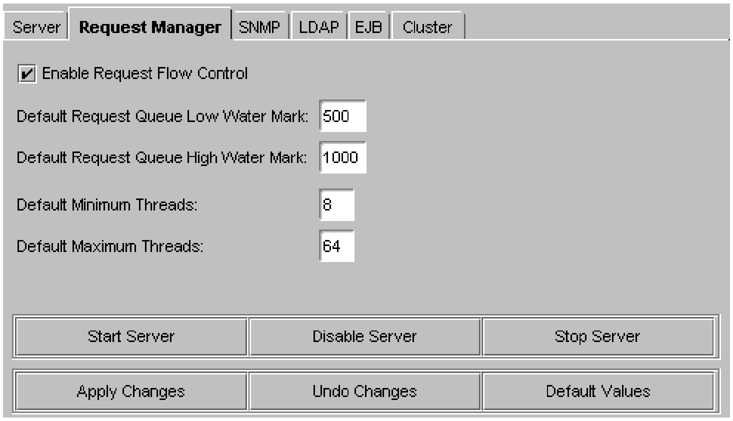
This section includes the following topics:

- To Control Request Flow at Server Level
- To Control Request Flow at Process Level

To Control Request Flow at Server Level

To control the flow of requests at the server level, perform the following tasks:

1. On the iASAT toolbar, click General to open the General window.
2. In the left pane of the General window, select the server in which you want to control request flow.
3. Click the Request Manager tab from the right pane of the General window.
4. Mark the Enable Request Flow Control checkbox to enable flow control.



☑ Enable Request Flow Control

Default Request Queue Low Water Mark: 500

Default Request Queue High Water Mark: 1000

Default Minimum Threads: 8

Default Maximum Threads: 64

5. In the Request Queue Low Water Mark text box specify the number of requests that should be available in the queue, which will trigger request logging.

   This number is only applicable after the maximum number of requests in the queue has been reached. See the next step.

6. In the Request Queue High Water Mark text box specify the maximum number of requests for the queue.

   When this number is reached no more user's requests will be accepted until the request queue reduces to the number specified as the low watermark.

7. Click Apply Changes to save your changes.

To Control Request Flow at Process Level

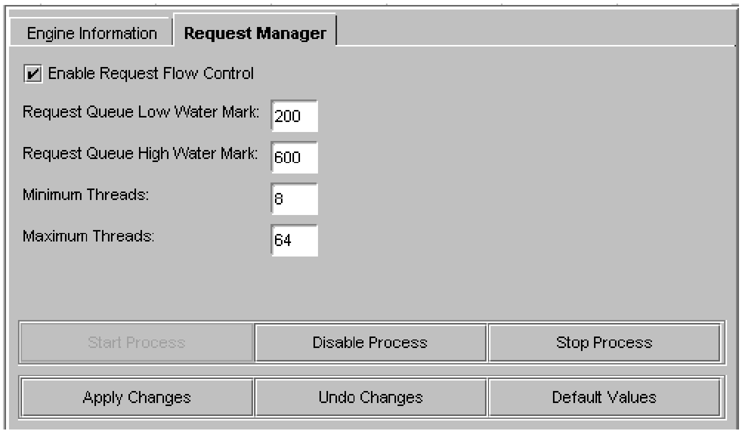
You can also customize the request flow for a process. Note that the numbers you specify for a process will be accepted as default for that process. The process level setting overrides the server level setting

To adjust the request flow for a process, perform the following tasks:

1. On the iASAT toolbar, click General to open the General window.
2. In the left pane of the General window, select the server for which you want to control request flow. Expand the server to view its processes.

NOTE You can expand a server in the hierarchical tree only when it is running.

3. Click the Request Manager tab in the right pane of the General window.
4. Click the Enable Request Flow Control checkbox to enable flow control.



☑ Enable Request Flow Control

Request Queue Low Water Mark: 200

Request Queue High Water Mark: 600

Minimum Threads: 8

Maximum Threads: 64

5. In the Request Queue Low Water Mark text box enter the number of requests in the queue in which logging will resume.

6. In the Request Queue High Water Mark text box enter the maximum number of requests for the queue.

   These setting override the default settings at the server level.

7. Click Apply Changes to save your changes.

NOTE Settings that are specified at process level override those set at the server level.

Setting Options of the Administration Server

The Administration Server (KAS) manages all the administrative processes and tasks within iPlanet Application Server. You can set several options for the administrative server that will enable high availability of server resources. Setting these options can increase the performance of applications running on a server and attempt to reduce the likelihood of application downtime. You can set the following options:

- To Specify EJB Container Parameters for Run Time
- To Specify Maximum Number of Engine Restarts
- To Enable Internationalization Support

- To Cache JavaServer Pages (JSP)
- To Specify Maximum Server and Engine Shutdown Time

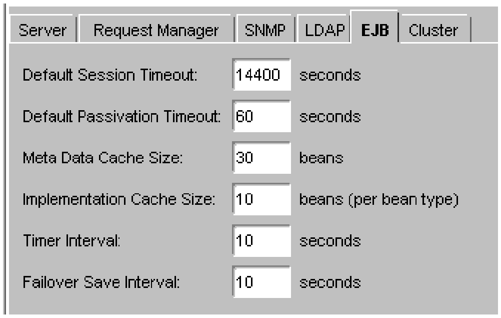
To Specify EJB Container Parameters for Run Time

iPlanet Application Server provides an EJB container that enables you to build distributed applications using your own EJB components, and components from other suppliers. When you configure iPlanet Application Server for your enterprise, you must set the EJB container's declarative parameters. These parameters determine, for example, session timeout when an EJB is removed after being inactive for a specified number of seconds. Set these parameters using the editor in iASAT.

To access the editor, perform the following tasks:

1. On the iASAT toolbar, click General to open the General window.
2. In the right pane of the General window, click the EJB tab to open the EJB container declarative parameters editor.

   The following window appears:



Default Session Timeout: 14400 seconds

Default Passivation Timeout: 60 seconds

Meta Data Cache Size: 30 beans

Implementation Cache Size: 10 beans (per bean type)

Timer Interval: 10 seconds

Failover Save Interval: 10 seconds

   The editor allows you to set the following values:

   ❍ Default Session Timeout

     If an EJB is not accessed for the specified number of seconds, it is removed. This applies to stateful session EJBs.

   ❍ Default Passivation Timeout

Time (in seconds) that elapses before the state of the EJB is written to disk. This value must be less than the session timeout value.

- Metadata Cache Size

  The metadata cache for EJBs, whose value is in number of EJBs.

- Implementation Cache Size

  Maximum cache size is in number of EJBs.

- Timer Interval

  How frequently (in seconds) the EJB pool checks to see if it should passivate or remove an EJB.

- Failover Save Interval

  How frequently (in seconds) the EJB state is saved. If the server fails, the last saved state of the EJB can be restored. Data saved is available to all engines in a cluster. This value is set on a per server basis and applies to EJBs that were deployed with Failover option enabled (on the General tab of the Deployment Tool EJB descriptor editor).

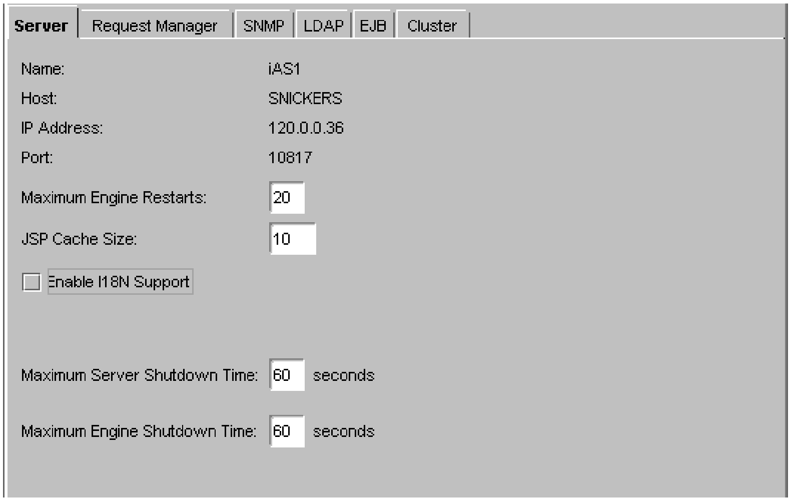
3. Click Apply Changes to save your changes.
4. Restart iPlanet Application Server for the changes to take effect.

To Specify Maximum Number of Engine Restarts

When a process, such as Executive Server (KXS), Java Server (KJS), C++ Server (KCS) or Corba Executive Server (CXS) fails, the Administrative Server restarts it. You can set the restart option to either increase or decrease the number of times that a process is restarted. Fault tolerance and application availability are increased when all processes are running smoothly.

To adjust the restart option of the Administrative Server, perform the following tasks:

1. On the iASAT toolbar, click General to open the General window.
2. In the left pane of the General window, select the iPlanet Application Server instance whose Administrative Server restart option you want to adjust.
3. Open the Server tab from the right pane of the General window.
4. Specify the new restart value in the Maximum Engine Restarts text field.



Name: iAS1
Host: SNICKERS
IP Address: 120.0.0.36
Port: 10817
Maximum Engine Restarts: 20
JSP Cache Size: 10
□ Enable I18N Support
Maximum Server Shutdown Time: 60 seconds
Maximum Engine Shutdown Time: 60 seconds

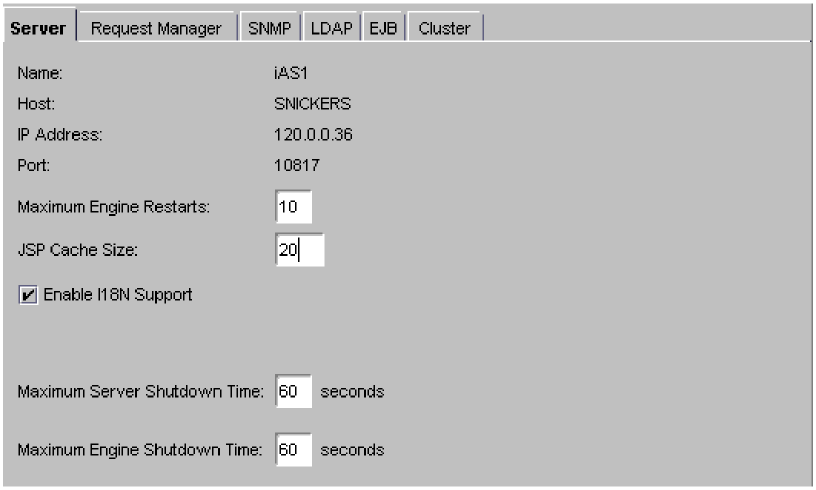
5. Click Apply Changes to save your changes.

To Enable Internationalization Support

You can enable iPlanet Application Server to support applications in languages belonging to different geographic locations.

To enable internationalization, perform the following tasks:

1. On the iASAT toolbar, click General to open the General window.
2. In the left pane of the General window, select the iPlanet Application Server instance for which you want to enable internationalization.
3. Click the Server tab in the right pane of the General window.
4. Mark the Enable I18N Support check box.



Name: iAS1

Host: SNICKERS

IP Address: 120.0.0.36

Port: 10817

Maximum Engine Restarts: 10

JSP Cache Size: 20

☑ Enable I18N Support

Maximum Server Shutdown Time: 60 seconds

Maximum Engine Shutdown Time: 60 seconds

5. Click Apply Changes to save your changes.

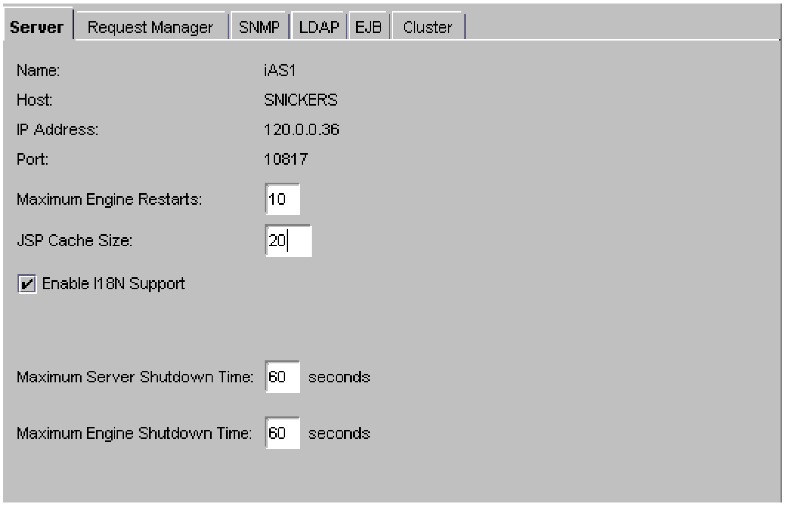
NOTE You must stop and restart the server for your changes to take affect. See Performing Administrative Tasks Using iPlanet Registry Editor for information.

To Cache JavaServer Pages (JSP)

You can specify the number of JSP pages that are cached by each KJS engine for each iPlanet Application Server instance. Caching JSPs optimizes application response time.

To set the JSP caching value of the Administrative Server, perform the following tasks:

1. On the iASAT toolbar, click General to open the General window.
2. In the left pane of the General window, select the iPlanet Application Server instance for which you want to set the JSP caching value.
3. Click the Server tab from the right pane of the General window.
4. Specify the JSP Cache Size in the text field. The cache size is set on a per-page basis.



Name: iAS1

Host: SNICKERS

IP Address: 120.0.0.36

Port: 10817

Maximum Engine Restarts: 10

JSP Cache Size: 20

☑ Enable I18N Support

Maximum Server Shutdown Time: 60 seconds

Maximum Engine Shutdown Time: 60 seconds

5. Click Apply Changes to save your changes.

To Specify Maximum Server and Engine Shutdown Time

You can set shutdown values of the Administrative Server for both iPlanet Application Server and engine processes. For example, if you set a 60 seconds engine shutdown time, application tasks being processed are allowed 60 seconds for completion. No new requests are accepted after this period has elapsed. Specifying a shutdown value avoids a “hard” shutdown that will return errors to the client.

To set the server and engine shutdown time of the Administrative Server, perform the following tasks:

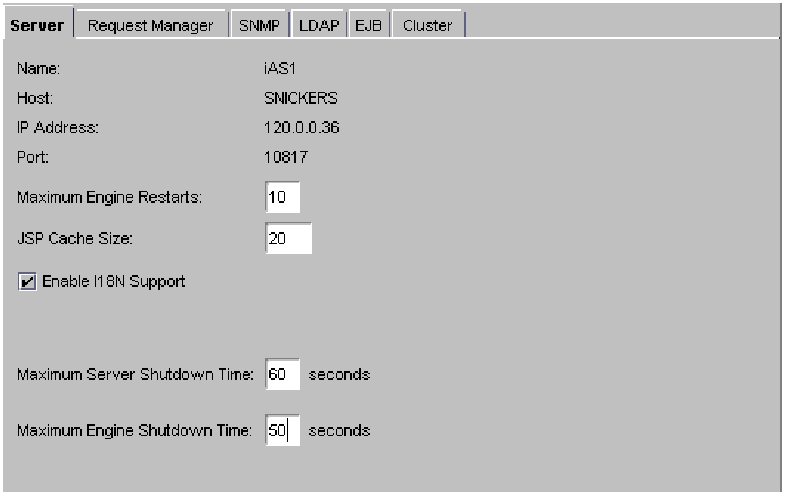
1. On the iASAT toolbar, click General to open the General window.
2. In the left pane of the General window, select the iPlanet Application Server instance whose shutdown time you want to specify.
3. Click the Server tab from the right pane of the General window.

   Specify a Maximum Server Shutdown Time.

The Maximum Server Shutdown Time is the maximum time taken to shut down iPlanet Application Server. After this time, any engines that are still running are killed. The server typically shuts down quickly unless it is heavily loaded.

4. Enter a Maximum Engine Shutdown Time.

   The Maximum Engine Shutdown Time is the maximum time that iPlanet Application Server will wait for an engine to shut down. After this time, the engine will be killed, and the next engine(s) will be shutdown.



Name: iAS1
Host: SNICKERS
IP Address: 120.0.0.36
Port: 10817
Maximum Engine Restarts: 10
JSP Cache Size: 20
☑ Enable I18N Support
Maximum Server Shutdown Time: 60 seconds
Maximum Engine Shutdown Time: 50 seconds

5. Click Apply Changes to save your changes.

Implementing a Multi-Process, Single-Threaded Environment

You can add a Java Server (KJS) or C++ Server (KCS) process to implement a multi-process, single-threaded environment. Running multiple KJS processes, all in single-threaded request mode, effectively creates a “multi-threaded” environment, which allows simultaneous processing of users requests.

Implementing this environment allows each process to accept only one request at a time. This is useful when you integrate third-party utilities. Running third-party utilities in the iPlanet Application Server multi-threaded request environment can cause errors that the application server may not be able to handle, including thread safety issues. You can implement a multi-process, single-threaded environment and avoid this type of problem, while enabling iPlanet Application Server to scale.

For example, if a third-party utility which is not thread safe runs within the KJS process, you can adjust the request threads of the KJS to 1 and eliminate the issues of utility safety. However, this creates a request backlog as requests wait for the KJS to process a single request at a time. You can avoid this by running multiple KJS processes in single-threaded request mode, and effectively create a "multi-threaded" environment allowing simultaneous processing of users' requests.

You do need to maintain multiple request threads for the Executive Server (KXS) process, as it distributes all requests that come into iPlanet Application Server.

To implement a multi-process, single-threaded environment, perform the following tasks:

1. Add a KJS or a KCS processes.

   See To Add and Tune Java and C++ Processes.

2. Adjust the request threads allocated for those processes to 1.

   See Adjusting the Number of Threads for Processing Requests.

Configuring Directory Server Failover

The Directory Server connected to your iPlanet Application Server instance contains global information shared by all application servers in a Directory Server cluster. A Directory Server cluster is simply one or more iPlanet Application Server instances that share a single Directory Server. See What Is LDAP? for more information on LDAP servers.

To protect this globally shared information, you need to configure a second Directory Server to act as a backup if the primary server fails. You need to install the backup Directory Server on a machine other than the one on which your primary Directory Server is configured.

Perform the following tasks to effectively configure a backup Directory Server for failover:

1. Install Netscape Directory Server 4.13, on a separate machine. See the documentation that is provided along with the server, for installation details.
2. Set up Supplier-Initiated Replication (SIR), for the backup Directory Server.
3. Modify the primary Directory Server configuration to include the backup Directory Server's hostname and port number. You can do this using iASAT.
4. If you have a webless installation of iPlanet Application Server, modify iPlanet Registry entries for the web-connector.

These topics are described in detail, in the following sections:

- Setting Up Supplier-Initiated Replication
- To Start Replication Agreement in Backup Directory Server
- To Set Up Replication of Directory Trees
- To Configure the Backup Directory Server in iASAT
- To Configure Registry Entries for Web Connector Plug-in

Setting Up Supplier-Initiated Replication

Supplier-Initiated Replication is a replication configuration where servers containing master copies of directory trees and sub-trees replicate directory data to replicated servers. To set up SIR, you need to first configure the backup Directory Server to accept replication configuration updates. You then need to set up replication agreements for specific directory trees in the primary Directory Server (see To Set Up Replication of Directory Trees)

Note that once the backup Directory Server is set up to accept replication configuration, any modifications made to the primary Directory Server will be replicated to the backup Directory Server.

To Start Replication Agreement in Backup Directory Server

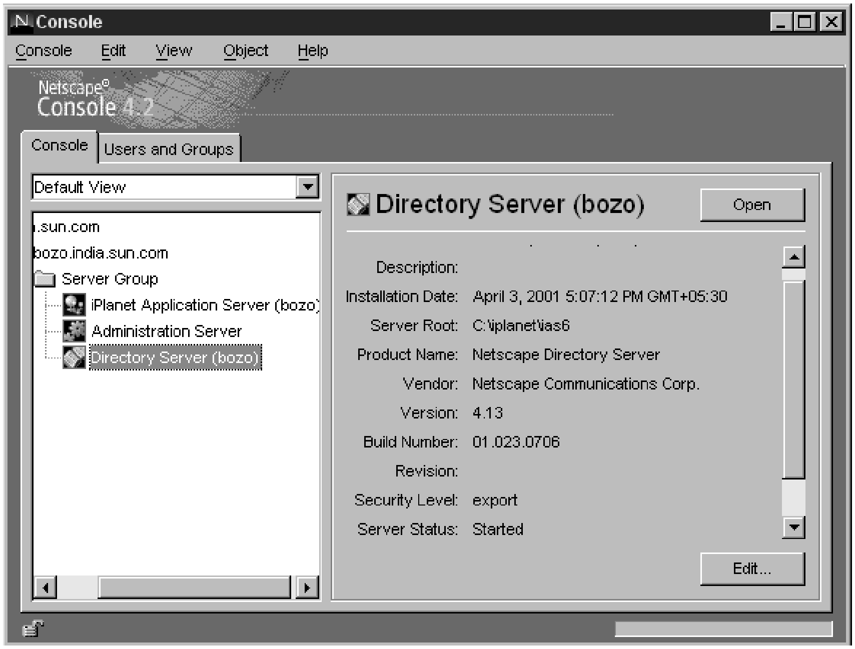
To configure the backup Directory Server to accept replication configuration updates, perform the following tasks:

1. Start Netscape Console (on the machine in which the backup Directory Server is installed).

On Windows systems, you can do this by opening the `Start` menu, and choosing `Programs>iPlanet Server Products>Netscape Console 6.0`.

On both Windows and Solaris systems, navigate to the `<iASInstallDir>` path, and type `startconsole` at the command line prompt.

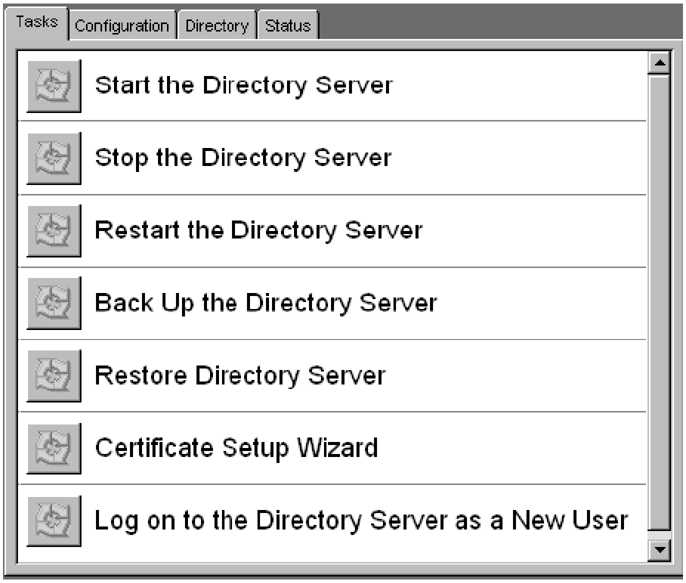
2. Double-click Directory Server, as shown in the following figure:



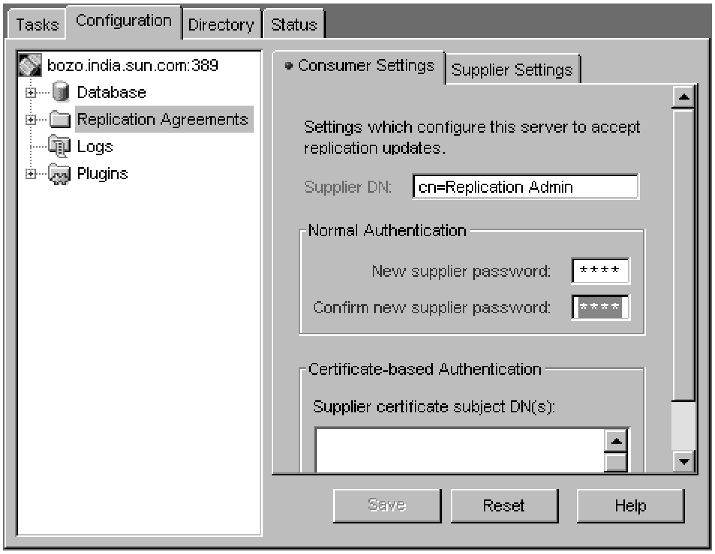
Directory Server (bozo)

Description:
Installation Date: April 3, 2001 5:07:12 PM GMT+05:30
Server Root: C:\iplanet\ias6
Product Name: Netscape Directory Server
Vendor: Netscape Communications Corp.
Version: 4.13
Build Number: 01.023.0706
Revision:
Security Level: export
Server Status: Started

The Netscape Directory Server opens, as shown in the following figure:


Tasks
Configuration
Directory
Status
Start the Directory Server
Stop the Directory Server
Restart the Directory Server
Back Up the Directory Server
Restore Directory Server
Certificate Setup Wizard
Log on to the Directory Server as a New User

3. Click the Configuration tab. The following window appears:



Settings which configure this server to accept replication updates.

Supplier DN: cn=Replication Admin

New supplier password: ****

Confirm new supplier password: ****

Supplier certificate subject DN(s):

4. Select the Replication Agreement folder as shown in the figure.

5. In the Consumer Settings tab, provide a valid Distinguished Name in the Supplier DN text field, for example, `cn=Replication Admin`.

6. In the New supplier password text field, enter a password of your choice. You will be prompted for this password during supplier authentication. Confirm your new password in the Confirm new supplier password text field. Note that your password must be at least 8 characters long.

   If you want certificate based authentication, enter the Distinguished Name of the supplier certificate that you want to use for authentication, in the Supplier certificate subject DN(s) text box.

NOTE For more information on certificate based authentication, see the chapter "Managing Replication," in the *Netscape Directory Server Administrator's Guide*. This document is installed with your installation of Directory Server in the following location:

`<iASInstallDir>/manual/en/slapd/ag/replicat.htm`

---

7. Click Save. The backup Directory Server is now ready for replication.

To Set Up Replication of Directory Trees

To configure the backup Directory Server for authentication, you need set up replication agreements for the following directory trees:

`ou=people, X`

`ou=Groups, X`

`ou=Y, o=NetscapeRoot`

The value of X is `YourDomainName` (for example, `india.sun.com`). This value can be specified by you during iPlanet Application Server installation. The value for Y is `iASCluster` (the clusters that are supported by the primary Directory Server). This value can't be changed.

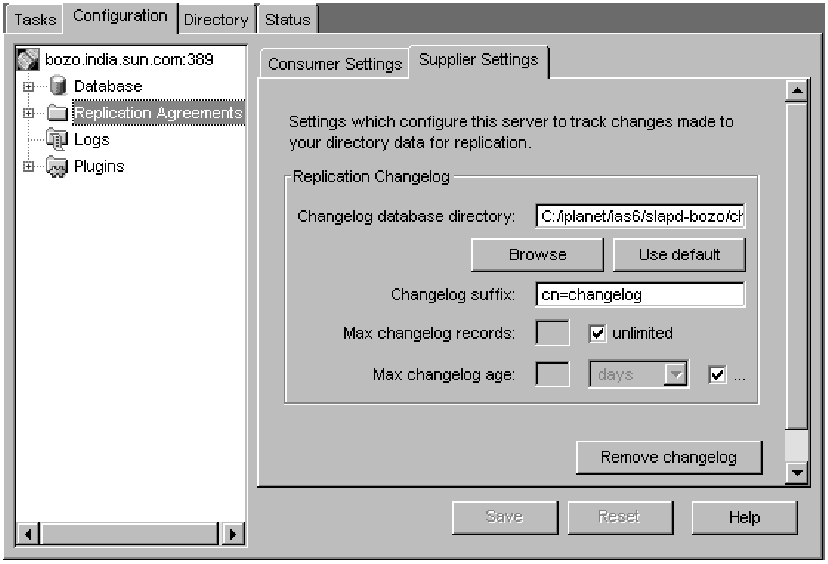
To set up directory tree replication, perform the following tasks:

1. Perform steps 1 to 3 as given in To Start Replication Agreement in Backup Directory Server.

2. Before you begin the replication, you need to set the primary Directory Server's changelog database directory. This is the directory where all the changes made to the directory server's settings are logged. You must set this directory to ensure that replication takes place.

   To set the changelog directory, perform the following tasks:

   a. In the Directory Server main window, open the Replication Agreements folder, as shown in the following figure:



Settings which configure this server to track changes made to your directory data for replication.

Replication Changelog

Changelog database directory: C:/iplanet/ias6/slapd-bozo/ch

Changelog suffix: cn=changelog

Max changelog records: ☑ unlimited

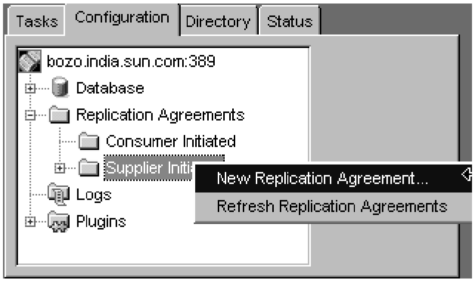
Max changelog age: days ☑ ...

   b. In the right pane of the window, click the Supplier Settings tab.

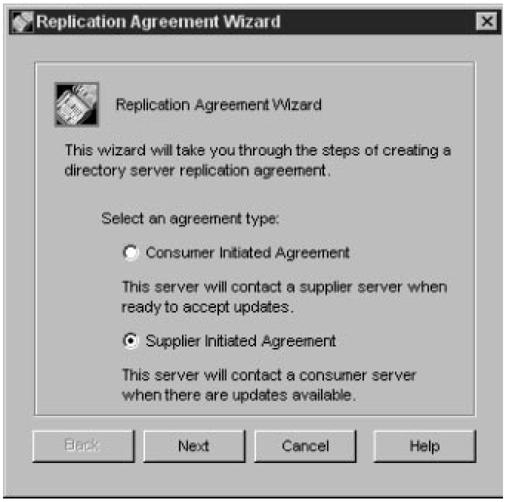
   c. Specify the changelog directory in the Changelog database directory text field. Click Browse to locate a directory, or click Use default to set the default changelog directory.

   d. Click Save to save your changes. You can now continue with the replication.

3. Expand the Replication Agreements folder to its hierarchal tree. Right-click on Supplier Initiated Agreements, and select New Replication Agreement, as shown in the following figure:



4. The Replication Agreement Wizard opens. Select Supplier Initiated Replication, as shown in the following figure:


Replication Agreement Wizard

Replication Agreement Wizard

This wizard will take you through the steps of creating a directory server replication agreement.

Select an agreement type:

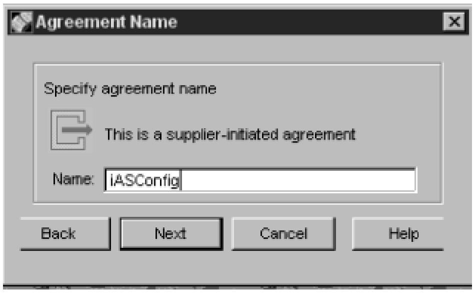
○ Consumer Initiated Agreement

This server will contact a supplier server when ready to accept updates.

● Supplier Initiated Agreement

This server will contact a consumer server when there are updates available.

5. Click Next. In the Agreement Name window, specify a name (any string value) for the supplier-initiated agreement, as shown in the following figure:



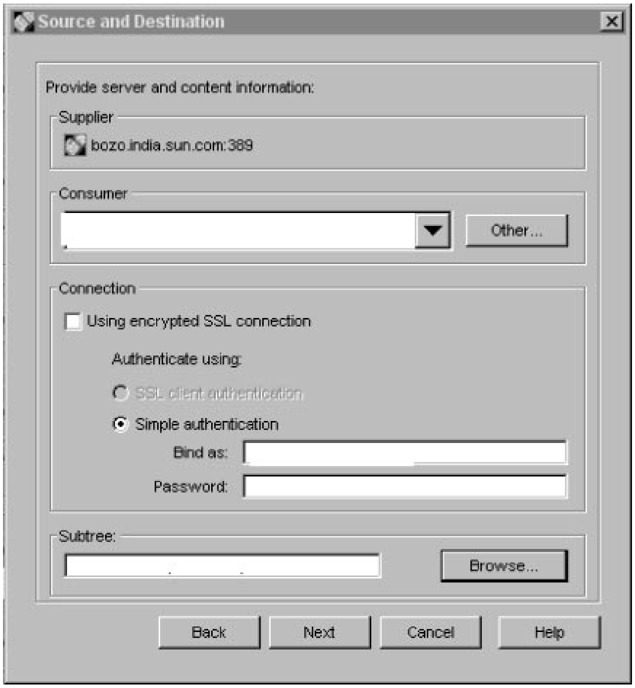
Agreement Name

Specify agreement name

This is a supplier-initiated agreement

Name: iASConfig

6. Click Next. The Source and Destination window opens, as shown in the following figure:



Source and Destination

Provide server and content information:

Supplier

bozo.india.sun.com:389

Consumer

Connection

☐ Using encrypted SSL connection

Authenticate using:

○ SSL client authentication

● Simple authentication

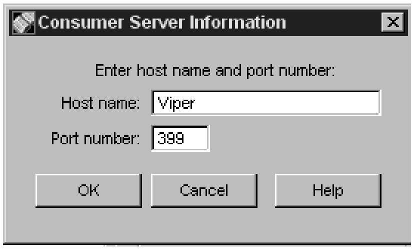
Bind as:

Password:

Subtree:

7. In the Consumer text field, enter the hostname of the machine on which the backup Directory Server is installed.

8. Click Other. The Consumer Server Information popup window opens, as shown in the following figure:



Consumer Server Information

Enter host name and port number:

Host name: Viper

Port number: 399

9. In the Host name text field, provide the host name of the machine on which the backup Directory Server is installed.

10. In the Port text field, provide the Port Number, if it is different from the primary Directory Server machine's port number.

11. Click OK to confirm. You can now see the consumer name in the Source and Destination window.

12. Select Simple authentication, for simple and direct authentication.

NOTE If you want a higher level of security, you can choose encrypted SSL connection. If you choose this option, the user will be authenticated through a Secure Socket Layer (SSL) process. You will need to specify the subject DN of the certificate server. For more information, see

---

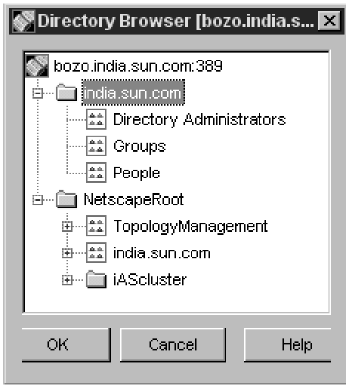
13. Provide a valid Distinguished Name in the Bind as text field, using which the Primary Server will bind to the backup Directory Server.

14. In the Password text field, provide the password that you used during iPlanet Application Server installation.

NOTE You can have multiple instances of iPlanet Application Server on a Solaris system. Ensure that the password you provide pertains to the iPlanet Application Server instance for which you are configuring a backup Directory Server.

---

15. In the Subtree field, provide the name of the directory tree that you want to replicate. Click Browse to view a list of the existing trees and subtrees in the primary Directory Server. The following window appears:



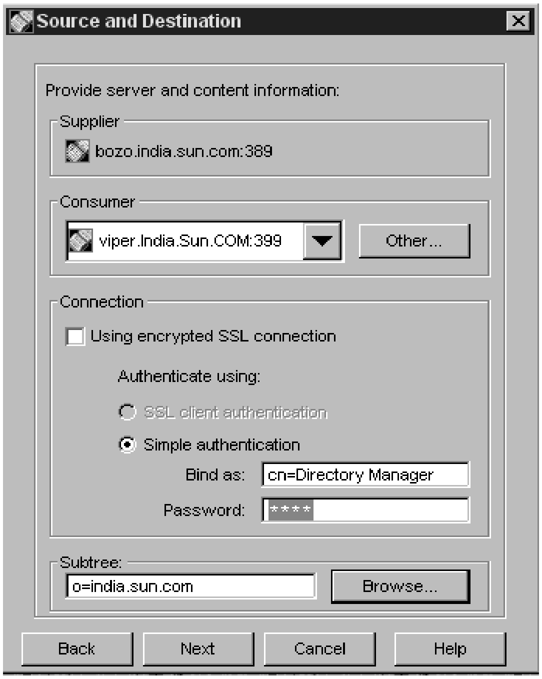
16. Select the first directory tree to be replicated. You can select either `X=YourDomainName` or `Y=iASCluster`. In the example, `X=YourDomainName` (`india.sun.com`) has been chosen.

    To choose `Y=iASCluster`, expand the `NetscapeRoot` folder and select `iASCluster`.

17. Click OK to confirm replication of the directory tree in the backup Directory Server.

NOTE When you replicate `X=YourDomainName`, make sure that you replicate all the domain names that are stored in your primary Directory Server, to enable proper authentication during failover.

The subtree that you selected appears in the Subtree text field of the Source and Destination window.



Source and Destination

Provide server and content information:

Supplier: bozo.india.sun.com:389

Consumer: viper.India.Sun.COM:399

Connection

☐ Using encrypted SSL connection

Authenticate using:

○ SSL client authentication

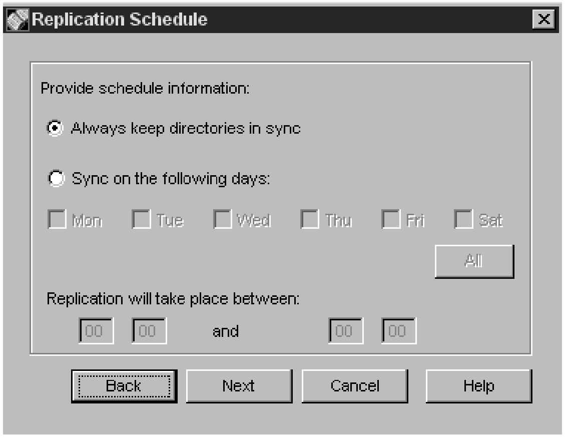
● Simple authentication

Bind as: cn=Directory Manager

Password: ****

Subtree: o=india.sun.com

18. Click Next. The Replication Schedule window appears, as shown in the following figure:



Replication Schedule

Provide schedule information:

● Always keep directories in sync

○ Sync on the following days:

☐ Mon ☐ Tue ☐ Wed ☐ Thu ☐ Fri ☐ Sat

Replication will take place between:

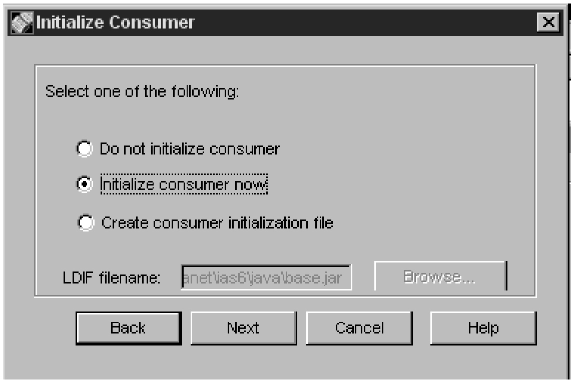
00 00 and 00 00

19. Select *Always keep directories in sync*, to keep the data current in both Directory Servers.

Select *Sync on the following days* if you want to schedule the data synchronization between the primary and backup Directory Servers. Data replication will occur during the period you specify on the selected days of the week.

Click All if you want to schedule data synchronization on all days of the week.

20. Click Next. The Initialize Consumer window appears, as shown in the following figure:



Initialize Consumer

Select one of the following:

○ Do not initialize consumer

● Initialize consumer now

○ Create consumer initialization file

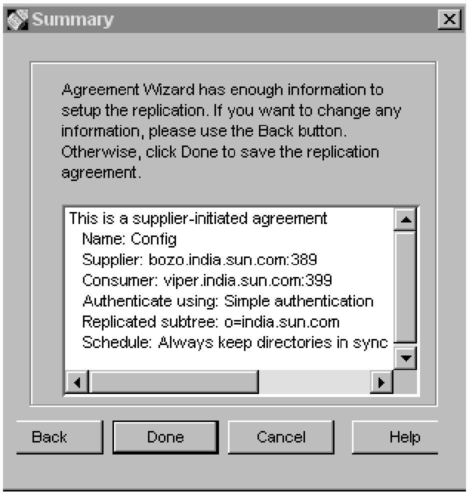
LDIF filename: anet\ias6\java\base.jar

21. Select the *Initialize consumer now* radio button. This option initializes the backup Directory Server immediately. The backup Directory Server is ready for replication.

Select *Do not initialize consumer* to stop replication at this point. If you choose this option, data is not replicated in the backup Directory Server.

Select *Create consumer initialization file* if you want to write the data to a file. Select a `.ldif` file for storing the data, which can be copied to the backup Directory Server later. See Using LDIF to Add Entries to Directory Server, for more information.

22. The Replication Summary window appears, as shown in the following figure:

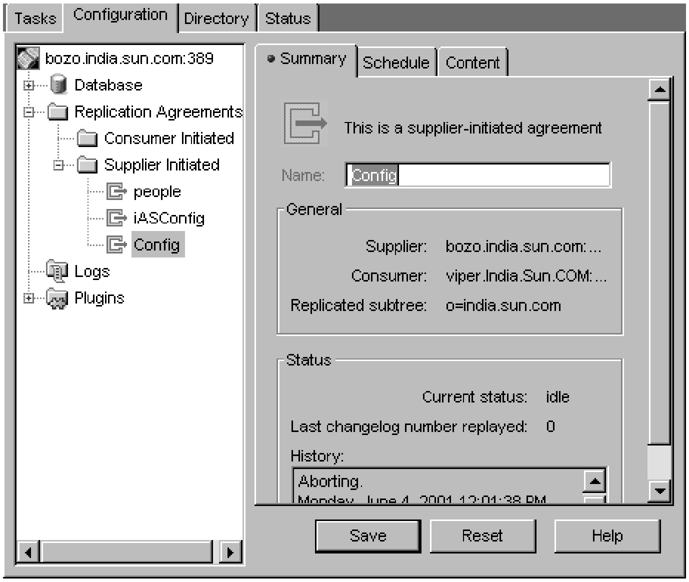


Agreement Wizard has enough information to setup the replication. If you want to change any information, please use the Back button. Otherwise, click Done to save the replication agreement.

```
This is a supplier-initiated agreement
  Name: Config
  Supplier: bozo.india.sun.com:389
  Consumer: viper.india.sun.com:399
  Authenticate using: Simple authentication
  Replicated subtree: o=india.sun.com
  Schedule: Always keep directories in sync
```

After viewing the summary, click Done. The replication agreement is created. The backup Directory Server is initialized and is populated with the primary Directory Server data.

You now need to replicate the remaining trees, `cn=iASCluster` and `o=NetscapeRoot`. Perform steps 1 to 18 and replicate these trees. When you complete the replication of all the required directory trees, the Replication Agreements folder should look like this:



This is a supplier-initiated agreement

Name: Config

Supplier: bozo.india.sun.com:...

Consumer: viper.India.Sun.COM:...

Replicated subtree: o=india.sun.com

Current status: idle

Last changelog number replayed: 0

History:

Aborting.

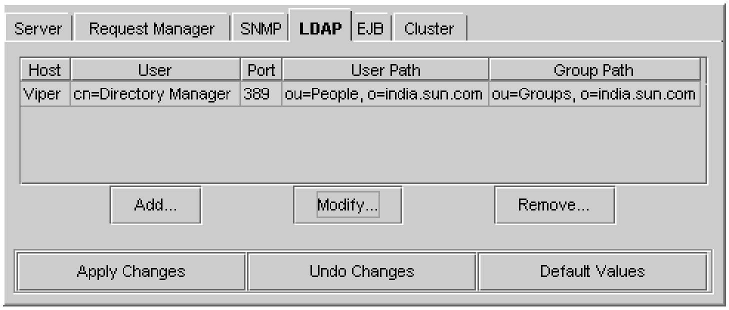
Monday, June 4, 2001 12:01:38 PM

Replication is ready to take place at this point.

To Configure the Backup Directory Server in iASAT

Once the backup Directory Server is installed and replicated, you need to modify the primary Directory Server configuration to include the name and port number of the backup Directory Server, in iASAT. You can do this by performing the following tasks:

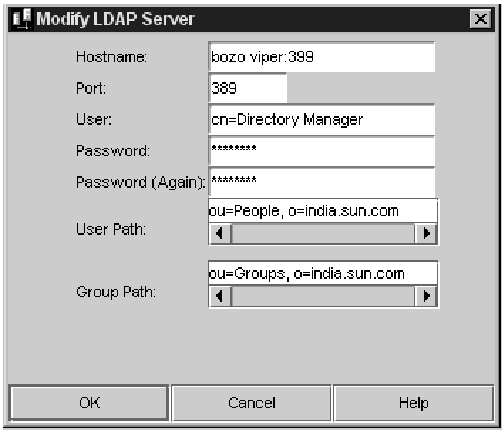
1. On the iASAT toolbar, click General to open the General window.
2. Select the iPlanet Application Server instance for which you want to configure a backup Directory Server.
3. Open the LDAP tab from the right pane of the General window. The following window appears:



Host	User	Port	User Path	Group Path
Viper	cn=Directory Manager	389	ou=People, o=india.sun.com	ou=Groups, o=india.sun.com

Directory Server(s) associated with your iPlanet Application Server instance appears in this window.

4. Select the primary Directory Server and click Modify. The following dialog box appears:



Modify LDAP Server

Hostname: bozo viper:399

Port: 389

User: cn=Directory Manager

Password: ********

Password (Again): ********

User Path: ou=People, o=india.sun.com

Group Path: ou=Groups, o=india.sun.com

5. In the Hostname field, insert a space after the existing hostname and enter the name of the machine on which the backup Directory Server is installed (as shown in the figure given above). If the port number of the backup Directory Server machine is different, insert a colon (`:`) next to the hostname and specify the port number, for example, `bozo viper:399`

   You need to specify the port number only if it is different from the port number of the host.

6. Click OK to close the window.

7. Click Apply Changes to register the changes in iASAT.

8. You need to stop and start iPlanet Application Server for the change to take effect.

The backup Directory Server is now configured. Note that you must always have at least one Directory Server configured to work with iPlanet Application Server.

NOTE To remove a Directory Server, select the required Directory Server in the right pane of the General window in iASAT, and click Remove.

To Configure Registry Entries for Web Connector Plug-in

When you configure a backup Directory Server, iPlanet Application Server creates the necessary back-end entries in iPlanet Registry. However, in the case of a webless installation of iPlanet Application Server, where iPlanet Application Server and the web-connector are installed on different machines, you have 2 instances of iPlanet Registry. You need to update iPlanet Registry entries on the web-connector, with the host name and port number of the machine on which the backup Directory Server is installed.

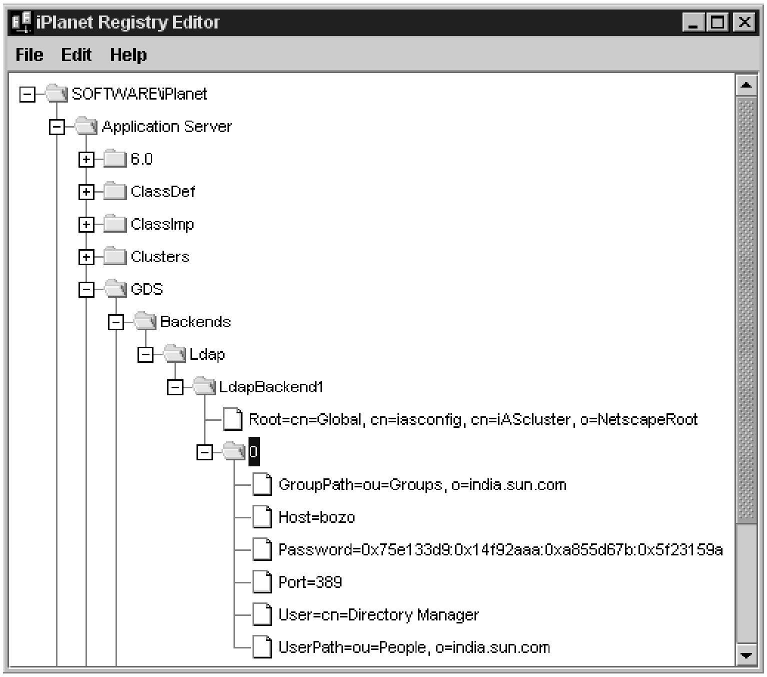
To update iPlanet Registry with the backup Directory Server details, perform the following tasks:

1. Start iPlanet Registry (on the machine where the web-connector is installed).

   (See About iPlanet Registry Editor).

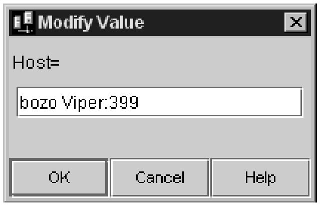
2. Open the following key:

   ```
   SOFTWARE/iPlanet/Application
   Server/GDS/Backends/Ldap/LdapBackend1/0/
   ```



```
SOFTWARE\iPlanet
  Application Server
    6.0
    ClassDef
    ClassImp
    Clusters
    GDS
      Backends
        Ldap
          LdapBackend1
            Root=cn=Global, cn=iasconfig, cn=iAScluster, o=NetscapeRoot
            0
              GroupPath=ou=Groups, o=india.sun.com
              Host=bozo
              Password=0x75e133d9:0x14f92aaa:0xa855d67b:0x5f23159a
              Port=389
              User=cn=Directory Manager
              UserPath=ou=People, o=india.sun.com
```

3. Select the `Host=<hostname>` value. Double-click the value to open it, or choose Modify Value from the Edit menu. The Modify Value dialog box appears, as shown in the following figure:



Host=

bozo Viper:399

4. Insert a space after the existing hostname and enter the name of the machine on which the backup Directory Server is installed (as shown in the figure). If the port number of the backup Directory Server machine is different, insert a colon `(:)` next to the hostname and specify the port number.

   For example, `bozo viper:399`

   Stop and start iPlanet Registry for the changes to take effect.

Chapter 7

Configuring the Web Connector Plug-In

This chapter describes the Web Connector Plug-in which sends users' requests to applications residing on iPlanet Application Server.

The following topics are included in this chapter:

- About the Web Connector Plug-In
- Manually Configuring a Web Server
- Configuring the Web Connector Plug-in for Web Server Logging
- Configuring Cookie and Hidden Field Usage
- Configuring a CGI Flag for CGI Requests
- Changing the Web Connector Port Number
- Specifying HTTP Variables for Input to Application Components

About the Web Connector Plug-In

The Web Connector Plug-in is installed on your web server at the time you install iPlanet Application Server.

If you install iPlanet Application Server on the same machine where a web server is installed, the Web Connector Plug-in is simultaneously installed and the web server configured automatically.

If you install iPlanet Application Server on a machine where a web server is not installed, you must manually install the Web Connector Plug-in on that web server machine. For more information about manually installing the web connector, see the *Installation Guide.*

You can configure the following Web Connector Plug-in functions:

Table 7-1 Configurable Web Connector Plug-in Functions

Connector functionality	Description	More information
Web server request logging	Mapping web server request components to database fields and adding HTTP variables to the log.	Configuring the Web Connector Plug-in for Web Server Logging
Cookie and hidden field security	Enable or disable cookies and hidden fields during web server to iPlanet Application Server communication.	Configuring Cookie and Hidden Field Usage
CGI flag for CGI request processing	Set a flag to process requests in CGI mode when that is necessary.	Configuring a CGI Flag for CGI Requests
The plug-in port number	Reconfigure the port number used by the plug-in.	Changing the Web Connector Port Number
Configuring HTTP variables as input for application components	Determine which HTTP variables can be accessed by application components.	Specifying HTTP Variables for Input to Application Components

Manually Configuring a Web Server

When you install iPlanet Application Server, your web server is automatically configured for the Web Connector Plug-in, meaning that all the necessary directories and settings on the web server are updated. However, there may be occasions, when, after you've installed the Web Connector Plug-in, you must manually re-configure the web server. This procedure is recommended only if you are having problems with the connection between iPlanet Application Server and your web server.

The following steps explain how to manually configure a web server to use the Web Connector Plug-in, whether your web server resides on the same or a different machine than where iPlanet Application Server is installed.

If you perform only step one of the following procedure (enabling CGI), the Web Connector Plug-in will run as a CGI script. If you perform the entire procedure, the Web Connector Plug-in will run as a plug-in, which is more efficient since a plug-in is faster than a CGI script.

You must be logged in as the same administrator user who installed the web server.

To Reconfigure an iPlanet Web Server

To reconfigure an iPlanet Web Server, perform the following steps:

1. Enable CGI, if it is not already enabled:
   a. On Windows, from the Start menu, select `iPlanet Web Server> Administer Web Servers`.
   b. Enter the administrator ID and password, and click OK.
   c. On the iPlanet Server Selector screen, choose the web server instance you want to configure from the drop-down list and click Manage.
   d. Click Programs, from the menu toolbar.
   e. On the CGI directory screen under URL prefix, type `cgi-bin`.
   f. Under CGI directory, enter the cgi-bin path.

   For iPlanet Web Server 4.1, Windows:

   ```
   drive letter:\Netscape\Server4\docs\cgi-bin
   ```

   For iPlanet Web Server 4.1, Solaris:

   ```
   <iASInstallDir>/docs/cgi-bin
   ```

   Now you are ready to configure the Web Connector Plug-in.

2. Edit the `obj.conf` file in the web server configuration directory.

   For iPlanet Web Server 4.1, Windows:

   ```
   drive letter:\Netscape\Server4\https-machinename\config
   ```

   For iPlanet Web Server 4.1, Solaris:

   ```
   iASInstallDir/https-machinename/config
   ```

   Make a copy of the file before modifying it. At the end of the `Init` section of the `obj.conf` file, add the following as two lines:

- Windows:

  ```
  Init fn="load-modules"
     funcs=nas_name_trans,gxrequest,gxlog,gxinit,gxredirect,
     gxhtmlrequest shlib="path to iAS bin dir/example:
     gxnsapi6.dll"
  ```

  ```
  Init fn="gxinit"
  ```

- Solaris:

  ```
  Init fn="load-modules"
     funcs=nas_name_trans,gxrequest,gxlog,gxinit,gxredirect,
     gxhtmlrequest shlib="libgxnsapi6.so"
  ```

  ```
  Init fn="gxinit"
  ```

Specify the following for `shlib`, iPlanet Enterprise Web Server 4.1:

- Windows:

  <iASInstallDir>`\bin\gxnsapi351l`

- Solaris:

  <iASInstallDir>`/gxlib/libgxnsapi30.so`

3. In the `Object name=default` section, just after `type=text/plain` section, add the following line:

   ```
   Service fn="gxredirect" fnname="imagemap" method="(GET|HEAD)"
   ```

4. In the `Object name=cgi` section(s), insert the following line immediately before the line `Service fn="send-cgi"`:

   ```
   Service fn="gxrequest"
   ```

   And then insert the following line immediately after the line `Service fn="send-cgi"`:

   ```
   AddLog fn="gxlog"
   ```

5. Make a copy of the current version of the file `obj.conf` and copy it to the back up version (so that the backup is consistent with the current version) in the following directory:

   For Windows:

   ```
   drive letter:\iPlanet\SuiteSpot\https-machinename\conf_bk
   ```

   For Solaris:

   ```
   iPlanet install directory/https-machinename/conf_bk
   ```

6. **Solaris only**: Modify the web server’s start and stop scripts as follows:

In the start script:

Set `GX_ROOTDIR` to the directory in which iPlanet Application Server is installed. For example:

```
GX_ROOTDIR=iASInstallDir; export GX_ROOTDIR
```

7. Restart the web server.

You can install and configure Apache Web Server 1.3.19 and use it with iPlanet Application Server's Web Connector Plug-in. See 'Configuring Apache Web Server' in the Installation Guide, to know more about installing and configuring Apache Web Servers.

To Reconfigure the Microsoft Internet Information Server

Keep in mind the following information when reconfiguring Microsoft IIS:

- Rename the `gxisapi.dll` library to `gx.dll` and leave it in the cgi-bin directory of the IIS wwwroot (`inetput/wwwroot/cgi-bin/`).
- Configure the ISAPI filter file, `gx.dll`, in the following registry entry:

  ```
  My Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
  \W3SVC\Parameters\
  ```

  A string key, `Filter DLLs`, should be added under Parameters, with the following value:

  ```
  c:\inetpub\wwwroot\cgi-bin\gx.dll
  ```

Configuring the Web Connector Plug-in for Web Server Logging

Web server requests are divided into components. Each component is represented by an HTTP variable. HTTP variables are standardized across all web servers, so the configurations you make with regard to their use are web- server independent.

This section describes the following topics:

- Mapping HTTP Variables to Database Fields
- Adding HTTP Variables to the Log

Mapping HTTP Variables to Database Fields

To enable logging of a particular component of a web server request, you must map HTTP variables to specific database fields to ensure that web server requests are properly logged. Mapping HTTP variables to database fields is done in the Web Connector Plug-in on the web server machine. The web server machine may or may not be the same machine where you installed iPlanet Application Server.

To map HTTP variables to database fields, perform the following steps:

1. Open the iPlanet Registry Editor by typing `kregedit` at the command line.

   The editor tool opens and displays the keys and values that apply to iPlanet Application Server. If the web server and iPlanet Application Server are installed on separate machines, the editor opens and displays the keys and values that apply to the Web Connector Plug-in.

2. Open the following key:

   ```
   SOFTWARE\iPlanet\Application Server\6.0\CCS0\HTTPLOG\INPUTVARS
   ```

   Each value under this key represents an HTTP variable and the database field to which the variable is mapped.

   The ID of the value is the HTTP variable. The string value is the database field.

   The HTTP variable is in ALL CAPS, such as `HTTP_REFERER`, and the database field is exactly as it appears in the database table.

3. Double-click the HTTP variable you want to map to a database field.

   The String editor dialog box appears.

4. Enter the database field name as the value data and click OK.
5. Leave any HTTP variables you do not want to log blank.
6. Close the editor.

See your web server documentation for an explanation of the HTTP variables.

Use the iPlanet Registry Editor to modify the Web Connector Plug-in.

Adding HTTP Variables to the Log

You can also modify the list of available HTTP variables, adding variables to the list to expand your logging options.

To add HTTP variables to the log, perform the following steps:

1. Open the iPlanet Registry Editor by typing `kregedit` at the command line.

   The editor opens and displays the keys and values that apply to iPlanet Application Server. If the web server and iPlanet Application Server are installed on separate machines, the editor opens and displays the keys and values that apply to the Web Connector Plug-in.

2. Open the following key:

   ```
   SOFTWARE\iPlanet\Application Server\6.0\CCS0\HTTPLOG\INPUTVARS
   ```

   Each value under this key represents an HTTP variable and the database field to which the variable is mapped.

   The ID of the value is the HTTP variable. The string value is the database field.

   The HTTP variable is in ALL CAPS, such as `HTTP_REFERER`, and the database field is exactly how it appears in the database table.

3. Add a new String value with the new HTTP variable name.

4. Click OK.

5. Repeat steps 3 through 5 for each new HTTP variable.

6. Close the editor.

See your web server documentation for a list and an explanation of all available HTTP variables.

Configuring Cookie and Hidden Field Usage

iPlanet Application Server is designed to work with web browsers in all modes of cookie and hidden-field security. There are three configurations you can set for the Web Connector Plug-in to support the various security modes. These configurations are described in the following table:

Table 7-2 Configurations to Support Security Modes

Cookie setting	Description
0	Cookies and hidden fields are passed back to the requesting web browser. This is the default setting.
1	Only hidden fields are passed back to the requesting web browser.
2	Only cookies are passed back to the requesting web browser.

To configure cookie and hidden field usage, perform the following steps:

1. Open the iPlanet Registry Editor by typing `kregedit` at the command line.

   The editor tool opens and displays the keys and values that apply to iPlanet Application Server. If the web server and iPlanet Application Server are installed on separate machines, the registry editor opens and displays the keys and values that apply to the Web Connector Plug-in.

2. Open the following key:

   ```
   SOFTWARE\iPlanet\Application Server\6.0\CCSO\HTTPAPI
   ```

3. Double-click the `NoCookie` DWORD value.

   The DWORD editor dialog box appears.

4. To disable cookies being passed to the web browser, change the value data to 1.

5. To disable hidden fields being passed to the web browser, change the value data to 2.

6. To enable both cookie and hidden fields, change the value data to 0.

7. When finished, close the editor.

Configuring a CGI Flag for CGI Requests

Some requests must be processed in CGI mode. You can set a flag in the Web Connector Plug-in to identify those requests.

To configure a CGI flag for CGI requests, perform the following steps:

1. Open the iPlanet Registry Editor by typing `kregedit` at the command line.

   The editor opens and displays the keys and values that apply to iPlanet Application Server. If the web server and iPlanet Application Server are installed on separate machines, the editor opens and displays the keys and values that apply to the Web Connector Plug-in.

2. Open the following key:

   ```
   SOFTWARE\iPlanet\Application Server\6.0\CCSO\HTTPAPI
   ```

3. Double-click the `AgentToken` String value.

   The String Editor dialog box appears.

4. For the value data, enter the flag that marks requests for CGI mode processing.

5. Click OK.
6. Close the editor.

Changing the Web Connector Port Number

In certain configurations, the web connector port number might conflict with another software package. You can reconfigure the connector port number to resolve this conflict.

To change the Web Connector Plug-in port number, perform the following steps:

1. Open the iPlanet Registry Editor. by typing `kregedit` at the command line.

   The editor opens and displays the keys and values that apply to iPlanet Application Server. If the web server and iPlanet Application Server are installed on separate machines, the editor opens and displays the keys and values that apply to the Web Connector Plug-in.

2. Open the following key:

   ```
   SOFTWARE\iPlanet\Application Server\6.0\CCSO\HTTPAPI
   ```

3. Double-click the `ListenPort` DWORD value and change the value data to an available port number.
4. Click OK.
5. Close the editor.

Specifying HTTP Variables for Input to Application Components

HTTP variables can be passed as part of the application request to application components like Enterprise Java Beans (EJBs). This allows the developer to determine certain information about the request and use that information when processing the request.

For example, the application might look at the `HTTP_REFERER` variable to determine where the request is coming from. This information might be used to present a more individualized greeting screen, or to keep statistics about where requests originate.

You edit entries in the registry to manage the HTTP variables. You can enable and disable them as desired. By default, iPlanet Web Server provides the following HTTP variables:

HTTPS
AUTH_USER
CLIENT_CERT
CONTENT_LENGTH
CONTENT_TYPE
HOST
HTTP_ACCEPT
HTTP_ACCEPT_CHARSET
HTTP_ACCEPT_ENCODING
HTTP_ACCEPT_LANGUAGE
HTTP_AUTHORIZATION
HTTP_CONNECTION
HTTP_COOKIE
HTTP_HOST
HTTP_IF_MODIFIED_SINCE
HTTP_REFERER
HTTP_USER_AGENT
HTTP_USER_DEFINED
HTTPS_KEYSIZE
HTTPS_SECRETKEYSIZE
PATH_INFO
PATH_TRANSLATED
QUERY
QUERY_STRING
REMOTE_ADDR
REMOTE_HOST
REMOTE_IDENT
REMOTE_USER
REQUEST_METHOD
SCRIPT_NAME
SERVER_PORT
SERVER_PROTOCOL
SERVER_SOFTWARE
SERVER_URL

To specify HTTP variables for input to application components, perform the following steps:

1. Open the iPlanet Registry Editor by typing `kregedit` at the command line.

   The editor opens and displays the keys and values that apply to iPlanet Application Server. If the web server and iPlanet Application Server are installed on separate machines, the editor opens and displays the keys and values that apply to the Web Connector Plug-in.

2. Open the appropriate key:

   - For iPlanet web servers, open the following key:

```
SOFTWARE\iPlanet\Application
Server\6.0\CCSO\HTTPAPI\INPUTNSAPI
```

- For Microsoft web servers, open the following key:

```
SOFTWARE\iPlanet\Application
Server\6.0\CCSO\HTTPAPI\INPUTNSAPI
```

Each value name shown represents an HTTP variable. The value determines whether the HTTP variable is passed to iPlanet Application Server with the application request. If the name's value is non-zero, the HTTP variable is passed to the iPlanet Application Server machine with the application request.

The name is created in ALL CAPS, such as `HTTP_REFERER`.

3. Add a name that is the HTTP variable name.
4. Double-click the new HTTP variable (name) and enter the one of the following as the value:
   - Enter a 0 to disable the HTTP variable.
   - Enter a 1 to enable the HTTP variable.

NOTE You can disable any of the default HTTP variables by adding the HTTP variable name and then setting the key name value to 0. For example, you could add ENTITY_HEADER and set its value to 1 and then add HTTP_REFERER (a HTTP variable provided by default) and set its value to 0 to disable it.

5. Click OK.
6. Repeat steps 4 through 6 for each HTTP variable you want to add/enable/disable.
7. Close the editor.

Administering Database Connectivity

iPlanet Application Server applications can access database(s), to add, retrieve, and modify data. This chapter describes how to configure data access drivers and apply settings to database connectivity parameters.

The following topics are included in this chapter:

- About Data Access Drivers
- Adjusting Database Connectivity Parameters
- Setting up Native and Third-Party JDBC drivers
- Configuring Resource Manager
- Configuring Data Sources
- To Delete a Data Source

About Data Access Drivers

iPlanet Application Server applications often require database access. Database access is achieved through a data access driver, software written either by the database vendor or a third-party vendor. The following types of data access drivers can be configured with iPlanet Application Server to provide database connectivity:

- Oracle
- DB2
- Informix
- Sybase

- MSSQL server (for NT)
- ODBC

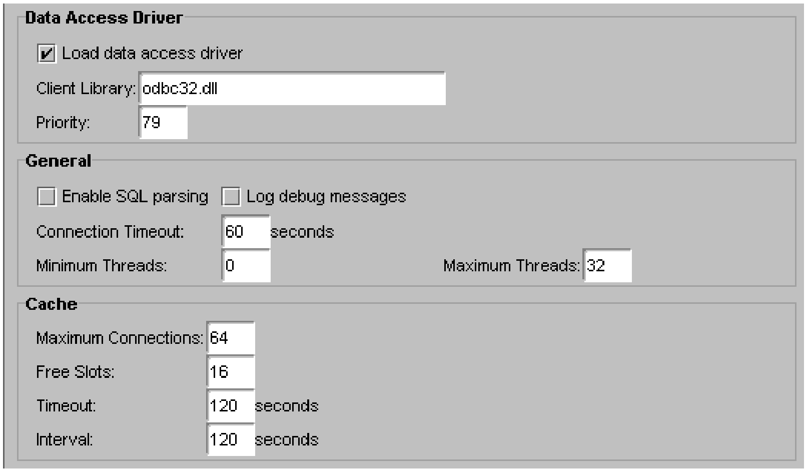
Make sure that data access drivers are installed before installing an instance of iPlanet Application Server. This way, iPlanet Application Server can automatically configure the drivers.

To Configure Data Access Drivers

When you open the Database window of iPlanet Administration Tool (iASAT), the left pane displays all data access drivers installed on a particular server whether the drivers are configured or not. A red X appears next to drivers that are not configured.

To configure a data access driver, perform the following steps:

1. From iASAT toolbar, click Database to open the Database window.
2. In the left pane of the Database window, select the driver you want to configure.
3. In the right pane of the Database window, mark the Load Data Access Driver checkbox, as shown in the following figure:



Data Access Driver

☑ Load data access driver

Client Library: odbc32.dll

Priority: 79

General

□ Enable SQL parsing □ Log debug messages

Connection Timeout: 60 seconds

Minimum Threads: 0 Maximum Threads: 32

Cache

Maximum Connections: 64

Free Slots: 16

Timeout: 120 seconds

Interval: 120 seconds

4. In the Client Library field, you can edit the library corresponding to the data access driver.
5. In the Priority field, you can edit the priority of the data access driver.

Giving a data access driver a priority of 1 means that driver has first priority over all other drivers. The higher the number, the lower the priority.

6. Click Apply Changes to save your changes to iPlanet Application Server.

7. Stop and start the server for the changes to take effect.

Adjusting Database Connectivity Parameters

iPlanet Application Server allows you to adjust database connectivity through connection parameters. Connection parameters allow you to optimize the speed with which iPlanet Application Server connects to a database or databases. The connection parameters are grouped in the following categories:

- connection
- threads
- result set buffer
- database cache

This section describes the following topics:

- To Set Connection Parameters
- To Set Thread Parameters
- To Set Database Cache Parameters

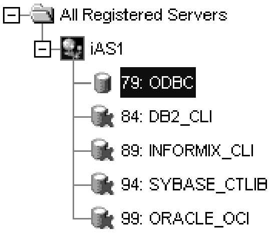
To Set Connection Parameters

You can set the length of time for which iPlanet Application Server attempts to make a database connection. These parameters optimize the performance of the iPlanet Application Server machine by keeping the server from wasting resources. For example, because iPlanet Application Server waits for open database connections when a request is made, the connection time limit is useful to limit the server from endlessly trying to connect to a database that is down.

To set the connection parameters, perform the following steps:

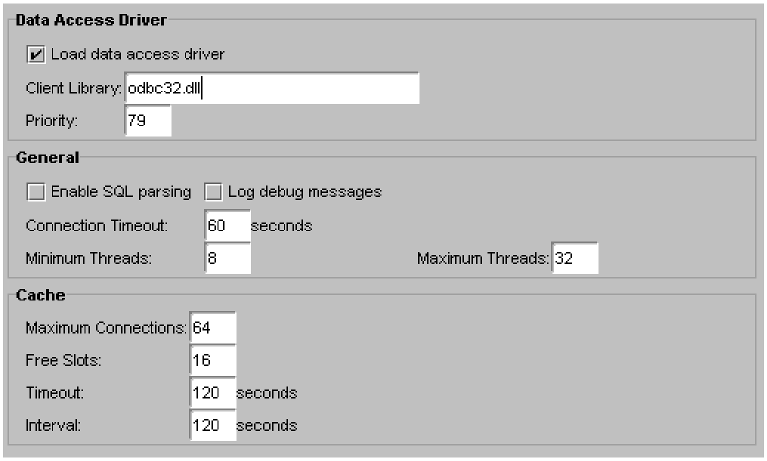
1. From the iASAT toolbar, click Database to open the Database window.

2. In the left pane of the Database window, click the database for which you want to adjust the timeout parameter, as shown in the following figure:



All Registered Servers
iAS1
79: ODBC
84: DB2_CLI
89: INFORMIX_CLI
94: SYBASE_CTLIB
99: ORACLE_OCI

3. In the right pane of the Database window, in the Connection Timeout field, enter the number of seconds.



Data Access Driver

☑ Load data access driver

Client Library: odbc32.dll

Priority: 79

General

☐ Enable SQL parsing ☐ Log debug messages

Connection Timeout: 60 seconds

Minimum Threads: 8 Maximum Threads: 32

Cache

Maximum Connections: 64

Free Slots: 16

Timeout: 120 seconds

Interval: 120 seconds

4. Click Apply Settings to save the changes to iPlanet Application Server.

To Set Thread Parameters

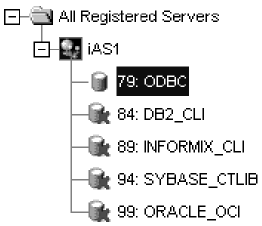
You can set the minimum and maximum number of threads available for database connections. The thread parameters determine how many threads iPlanet Application Server allocates for asynchronous database queries. Such threads are usually used for queries returning a large number of rows and allowing the application to do other tasks while waiting for the query to finish. Asynchronous database queries are not supported by JDBC 2.0, a Java programming interface used to build on top on database drivers.

The default thread allocations are adequate for most applications. If an application developer uses many asynchronous queries, you might want to increase the maximum number of available threads.

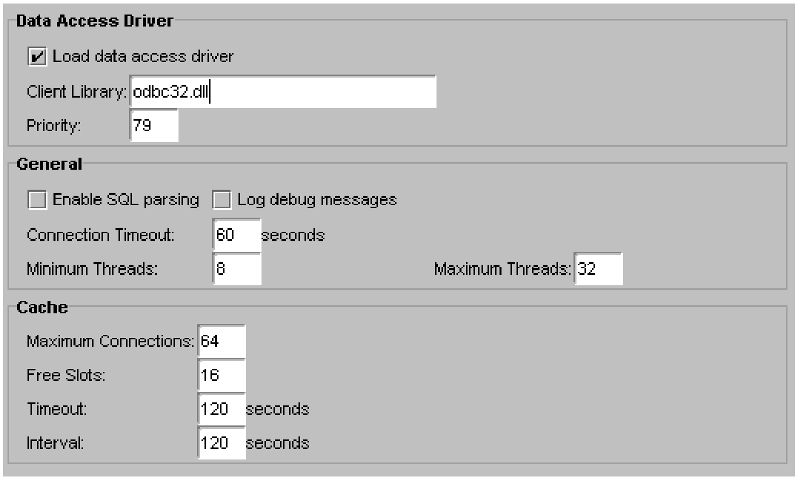
NOTE Each thread does use a small stack allocation and pulls from the total number of available system threads. Therefore, if an application does not use any asynchronous queries, you can increase performance by setting the maximum available threads to zero.

To set the thread parameters, perform the following steps:

1. From the iASAT toolbar, click Database to open the Database window.
2. In the left pane of the Database window, select the database for which you want to adjust the asynchronous thread parameters.



3. In the right pane of the Database window, in the Minimum Threads field, enter the number of threads.



Data Access Driver

☑ Load data access driver

Client Library: odbc32.dll

Priority: 79

General

□ Enable SQL parsing □ Log debug messages

Connection Timeout: 60 seconds

Minimum Threads: 8 Maximum Threads: 32

Cache

Maximum Connections: 64

Free Slots: 16

Timeout: 120 seconds

Interval: 120 seconds

4. In the right pane of the Database window, in the Maximum Threads field, enter the number of threads.
5. Click Apply Settings to save the changes to iPlanet Application Server.

To Set Database Cache Parameters

The database cache is an array used to hold active and recently used database connections. iPlanet Application Server adds database connections to cache when an application creates a database connection.

While the application is using that database connection, iPlanet Application Server marks that connection “in use.” Once the database operations are completed, the server marks the database connection “free.” The cache then holds the free connection in the cache for a configured period of time. This allows the server to use the free cached connection and quickly handle a new request to the same database. Once a free connection exceeds the timeout, a cleaning thread removes the connection from the cache and opens a slot for a new connection to be cached.

You can adjust the following cache parameters:

- maximum number of connections allowed in the cache
- number of slots held solely for free connections
- timeout limit, in seconds, for free connections

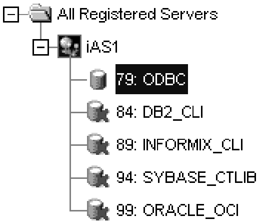
- interval, in seconds, at which the cache cleaner thread removes timed-out free connections

The default values are adequate for most applications, so adjustments are not usually required for initial application installations.

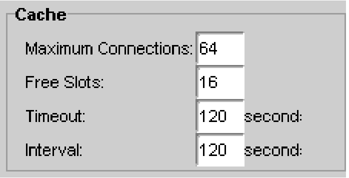
iPlanet Application Server dynamically adjusts the cache up to the maximum number of allowable connections. If there are no connections to cache, the array is allocated to zero spaces.

To set database cache parameters, perform the following steps:

1. From iASAT toolbar, click Database to open the Database window.
2. In the left pane of the Database window, select the database for which you want to adjust the database cache parameters.



3. In the right pane of the Database window, under Cache, enter values for the following parameters:
   - ❍ Maximum Connections
   - ❍ Free Slots
   - ❍ Timeout
   - ❍ Interval



Cache

Maximum Connections: 64

Free Slots: 16

Timeout: 120 second:

Interval: 120 second:

4. Click Apply Settings to save the changes to iPlanet Application Server.

Setting up Native and Third-Party JDBC drivers

JDBC is the Javasoft specification of a standard API that allows Java programs to access Database Management Systems (DBMSs). The JDBC API consists of a set of interfaces and classes that can be used to perform the following procedures:

- Write applications and applets that connect to databases.
- Send queries written in Structured Query Language (SQL).
- Process the results.

You can configure native and third party JDBC drivers when you install iPlanet Application Server. If you want to do this after installation, you can do so using a command line tool.

The following sections describe how to configure native and third party JDBC Drivers on both Solaris and Windows machines:

To Set Up Native JDBC Drivers on Solaris

To Set Up Third-Party JDBC Drivers on Solaris

To Set Up Native or Third-Party JDBC Drivers on Windows

To Set Up Native JDBC Drivers on Solaris

To configure native JDBC drivers on Solaris, perform the following steps:

1. Go to `<iasinstall>/ias/bin`. Run the following script:

   ```
   dbsetup_sh
   ```

   The setup prompts you to enter the following information. Follow the steps to set up native JDBC drivers.

2. Specify the directory in your machine on in which iPlanet Application Server has been installed, for example, `/iplanet`. Press Enter.

3. Choose the driver you want to install. You can see the following options:

   ```
   1. iPlanet Type 2 JDBC Drivers
   2. Third Party JDBC Drivers
   ```

   Type (1) to configure native JDBC drivers, and press Enter.

4. You are prompted to configure iPlanet Application Server with Oracle connectivity. Press ‘y’ if you want to configure Oracle connectivity.

Press ‘n’ to connect to any other database, such as Sybase, Informix, DB2. You can configure more than 1 database using the `dbsetup.sh` command line tool.

5. Specify the database home directory on your machine. If you choose to configure iPlanet Application Server with Oracle connectivity, specify the Oracle home directory, for example, `/usr/oraclient`.
6. You are then prompted to configure resource manager. Resource managers are required to manage global transactions (sometimes referred to as distributed transactions). To configure resource manager, type `Y` and press Enter.
7. Specify a name for resource manager, for example, RM1. You can configure more than 1 resource manager for a driver.
8. Enter database type, for example, `Oracle`.
9. Provide the default names for the following, as indicated:

   Database Server Name: `ksample`

   Database User Name: `kdemo`

   Database User Password: `kdemo`

   These defaults will be used to construct open string formats for different types of databases. These default values are present in the `TSNnames.ora` file, which can be found in the home directory of the oracle client installation.

   For more information on database open string formats, refer to the Installation Guide.

To Set up Third Party JDBC Drivers on Solaris

To configure third party JDBC drivers, perform Steps 1 and 2 as described in To Set Up Native JDBC Drivers on Solaris. In Step 3, choose the second option, that is, Third Party JDBC Drivers.

1. Specify the number of drivers you want to configure and press Enter.

   On Windows machines, you can configure a maximum of 3 third party JDBC drivers. These drivers can be configured for the same or different database clients.
2. Enter driver class name, for example `Database.Jdbc.driver.DatabaseDriver`. If your database is Oracle, type `Oracle.Jdbc.driver.OracleDriver`.

3. Enter the Driver class path, for example `JDBC/LIB/Classes/Classes111.zip.` This zip file holds the library classes for the driver. Specify the complete path as shown in the following example:

   `usr/oraclient/jdbc/classes/lib/classes12.zip.`

4. Specify the native driver directory for example, `usr/oraclient/lib`.

   Specify the native driver directory if you have installed third party JDBC driver Type 2. This entry is optional for JDBC drivers type 1, 3 or 4. For more information on driver types, visit `java.sun.com`.

   You are prompted to configure 2 more drivers. You can choose to do so, or press `Ctrl+C` to abort the process.

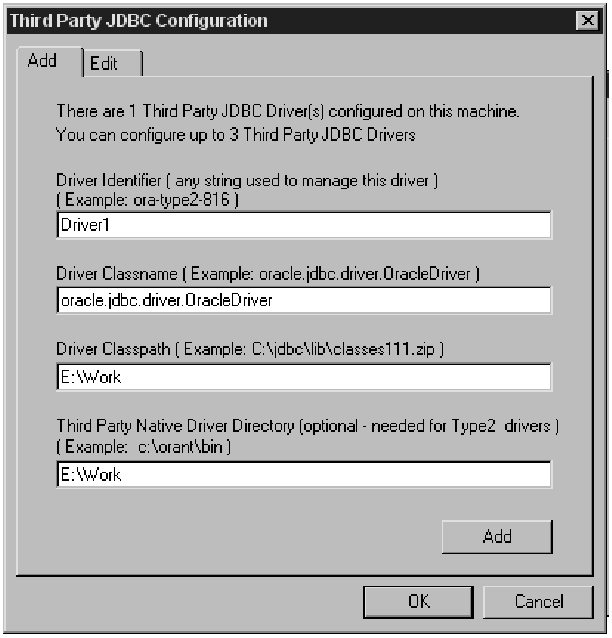
To Set Up Native or Third-Party JDBC Drivers on Windows

Perform the following steps to set up native or third party JDBC drivers on a Windows system:

1. Navigate to the `<iASinstall>/ias6/ias/bin` path. Run the following script:

   `JDBCsetup.exe`.

   The following window appears:



There are 1 Third Party JDBC Driver(s) configured on this machine.
You can configure up to 3 Third Party JDBC Drivers

Driver Identifier (any string used to manage this driver)
(Example: ora-type2-816)
Driver1

Driver Classname (Example: oracle.jdbc.driver.OracleDriver)
oracle.jdbc.driver.OracleDriver

Driver Classpath (Example: C:\jdbc\lib\classes111.zip)
E:\Work

Third Party Native Driver Directory (optional - needed for Type2 drivers)
(Example: c:\orant\bin)
E:\Work

2. In the Add tab of the Third Party JDBC Configuration window, specify a name for the driver, for example `Driver1`.

3. Enter driver classname, for example `Database.Jdbc.driver.DatabaseDriver`. If your database is Oracle, type `Oracle.Jdbc.driver.OracleDriver`. You cannot leave this field blank.

4. Enter the Driver class path, for example `JDBC/LIB/Classes/Classes111.zip.` This zip file holds the library classes for the driver. Specify the complete path as shown in the following example:

   `usr/oraclient/jdbc/classes/lib/classes12.zip.`

NOTE If you are setting up a native JDBC driver, do not specify this class path.

Specify the native driver directory for example, `usr/oraclient/bin`.

If you have installed third party JDBC driver Type 2, you need to specify this directory. This entry is optional for third party JDBC drivers Type 1, 3 or 4.

5. Click Add to register the driver.

   You are prompted to configure 2 more drivers. You can choose to do so, or click Cancel to abort the process.

Modifying JDBC Driver Settings. Open the Edit tab of the Third Party JDBC Configuration window. Click Update after modifying the settings.

Deleting a JDBC Driver. Open the Edit tab of the Third Party JDBC Configuration window. Select the required driver from the Driver Identifier field. Click Delete. The driver is removed from the registry.

Configuring Resource Manager

You need resource manager(s) to manage multiple databases and global transactions. Resource manager lets you connect to the relevant database back-end for global transactions. You need to configure a resource manager for each database back-end to which you want to connect. You can configure a maximum of 32 resource managers on your machine.

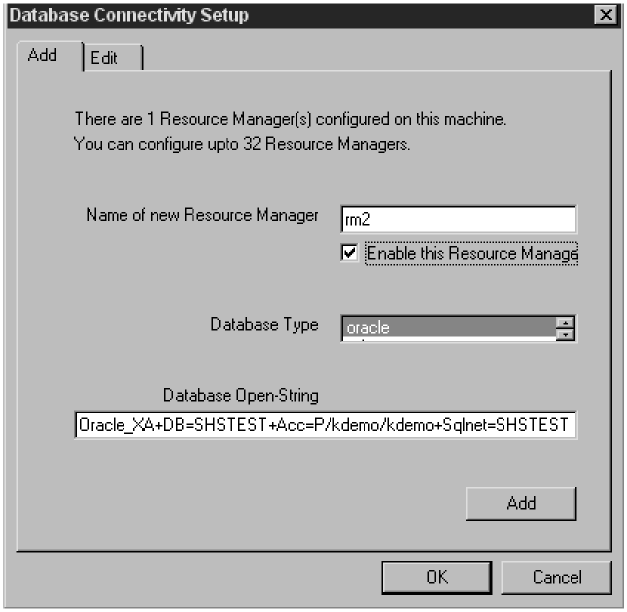
To Configure Resource Manager

To configure resource manager on Windows, perform the following steps:

1. Go to `<iASinstall>/ias6/ias/bin`. Run the following script:

   `DBsetup.exe`.

   The following window appears:



There are 1 Resource Manager(s) configured on this machine.
You can configure upto 32 Resource Managers.

Name of new Resource Manager: rm2

☑ Enable this Resource Manage

Database Type: oracle

Database Open-String: Oracle_XA+DB=SHSTEST+Acc=P/kdemo/kdemo+Sqlnet=SHSTEST

2. In the Add tab of the Database Connectivity Setup window, specify a name for a resource manager, for example, `RM1`.

3. Mark the checkbox Enable this Resource Manager, to use this resource manager to manage your database(s).

4. Select Database Type.

5. Enter the Database Open-String. This string provides information such as database name, host name, login ID, password, session time, login directory etc., to the resource manager. Refer to the *Installation Guide* for more information on string syntax.

6. Click Add to add the resource manager to iPlanet Registry.

Modifying Resource Manager Settings. Open the Edit tab of the Database Connectivity Setup window. Click Update after modifying the settings.

Deleting a Resource Manager. Open the Edit tab of the Database Connectivity Setup window. Select the required resource manager from the Name of new Resource Manager field. Click Delete. The driver is removed from the registry.

Configuring Data Sources

A data source contains information pertaining to the database client that the database driver needs to know, to connect to the database. Before you add a data source, you need to install and configure the database driver(s) on your machine. See Setting up Native and Third-Party JDBC drivers, for more information on setting up database drivers.

The following sections explain how you can add a data source, to both native and external database drivers:

- To Add a Data Source To a Native Database Driver
- To Add a Data Source to an External Database Driver

To Add a Data Source To a Native Database Driver

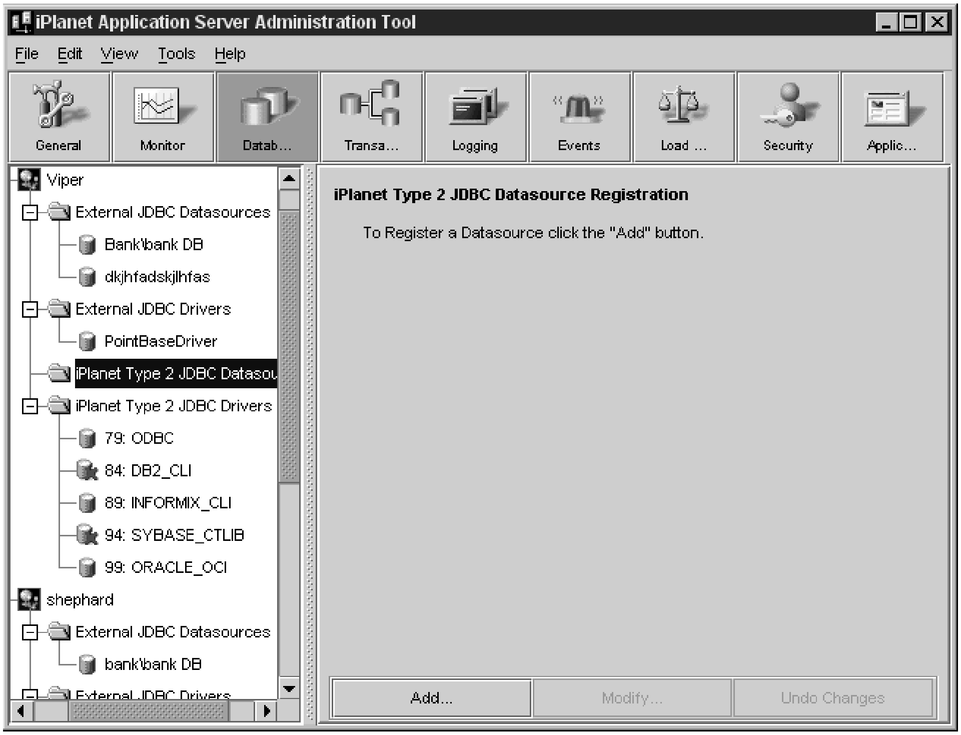
To add a data source to a Native database driver, such as a Type 2 JDBC driver, perform the following tasks:

1. Start iASAT. On Windows, from the Start menu, choose `Programs>iPlanet Application Server 6.0>`iPlanet Application Server `Administration Tool`.

   On Solaris, navigate to `<iASInstallDir>/ias/bin/` and type the following command:

   ```
   kvsradmin
   ```

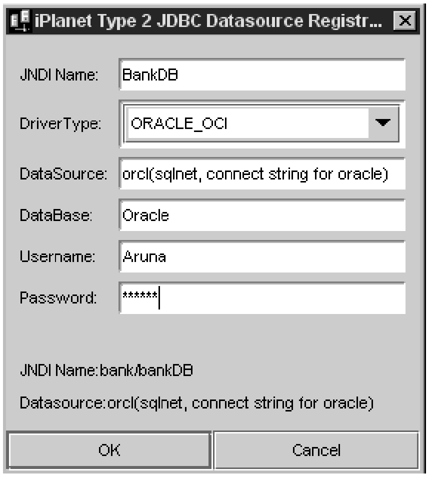
2. In the iASAT toolbar, click Databases.
3. In the left pane of the window, select the iPlanet Application Server instance for which you want to add a data source. You can see all the database connectivity settings that have been set for that machine.
4. Select the iPlanet Type 2 JDBC Datasources folder, as shown in the following figure:



iPlanet Type 2 JDBC Datasource Registration

To Register a Datasource click the "Add" button.

5. To add a data source, click Add in the right pane of the Databases window. The iPlanet Type 2 JDBC Datasource Registration window appears, as shown in the following figure:



JNDI Name: BankDB

DriverType: ORACLE_OCI

DataSource: orcl(sqlnet, connect string for oracle)

DataBase: Oracle

Username: Aruna

Password: ******

JNDI Name:bank/bankDB

Datasource:orcl(sqlnet, connect string for oracle)

6. In the JNDI name field, enter a name for the data source, for example, `BankDB`. When you register the data source, a node is created with this name.

NOTE The JNDI name should be unique for each data source that you add

7. In the Driver Type drop-down list, select the driver for which you want to add a data source.

NOTE For a native driver, you need to install the database client on your machine

8. In the DataSource field, enter the relevant value depending on the database client for which you are adding a data source. The database connects to the driver using this value. The following table provides the values for each database type that iPlanet Application Server supports:

Table 1 List of Data Source Values

Database Type	Value for the DataSource field
Oracle	TSN name of the Oracle database. You can find this value in the tsnnames.ora file which is present in the `ORACLE_HOME/network/admin` directory.

Table 1 List of Data Source Values

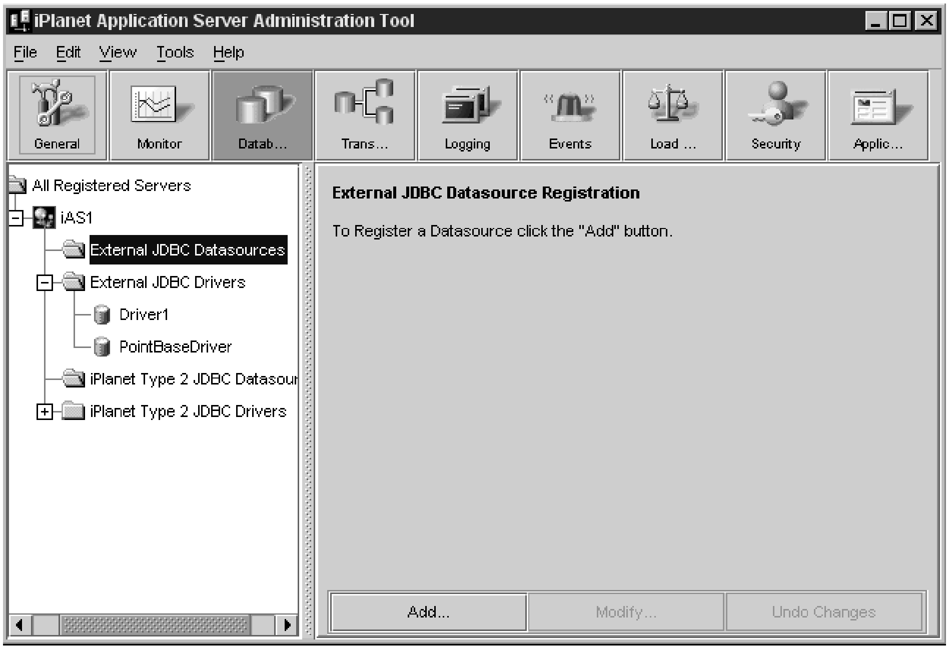
Database Type	Value for the DataSource field
Sybase	This is the name of the server on which the Sybase client has been installed.
DB2	Any string value. When you connect to the database, you need to provide this string value along with your user name and password.
Informix	You do not need to provide any value in this field. The Informix client automatically picks up the datasource when you log on.

9. In the DataBase field, provide the name of the database, for example, Oracle.
10. In the Username field, enter the user name that was provided when the database client was installed.
11. In the Password field, enter the password for that user name.
12. Click OK to register the data source. You need to stop and start iPlanet Application Server, for the changes to take effect. See Performing Administrative Tasks Using iASAT, for more information on starting and stopping iPlanet Application Server.

To Add a Data Source to an External Database Driver

To add a data source to an external database driver, such as a Type 4 JDBC driver, perform the following tasks:

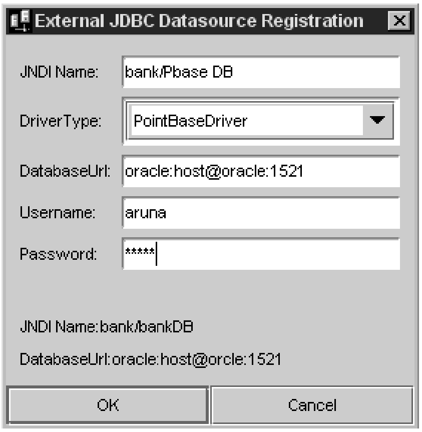
1. Start iASAT.
2. In the iASAT toolbar, click Databases. You can see all the database connectivity settings that have been set for that machine.
3. Select the External JDBC Datasources folder, as shown in the following figure:



External JDBC Datasource Registration

To Register a Datasource click the "Add" button.

4. To add a data source, click Add in the right pane of the Databases window. The Datasource Registration window appears, as shown in the following figure:



JNDI Name: bank/Pbase DB

DriverType: PointBaseDriver

DatabaseUrl: oracle:host@oracle:1521

Username: aruna

Password: *****

JNDI Name:bank/bankDB

DatabaseUrl:oracle:host@orcle:1521

5. In the JNDI name field, enter a name for the data source, for example, `Bank/PbaseDB`. When you register the data source, a node is created in iASAT with this name.

NOTE The JNDI name should be unique, for each data source that you add.

6. From the Driver Type drop-down list, select the driver for which you want to add a data source.

NOTE Type 4 drivers, you do not need to install the database client on your machine or server.

In the DatabaseURL field, provide the URL that points to the machine or server on which the database client has been installed. If the database client is installed on a machine which is connected to a different server, you need to mention the port number of this machine along with the URL.

For example, `oracle:host@oracle:1521`.

7. In the Username field, enter a user name that is registered in the database.
8. In the Password field, enter the password for that user name.
9. Click OK to register the data source. You need to stop and start iPlanet Application Server, for the changes to take effect. See Performing Administrative Tasks Using iASAT for more information on starting and stopping iPlanet Application Server.

To Delete a Data Source

To delete a data source, you need to remove the relevant entries from iPlanet Registry. To delete a data source, perform the following steps:

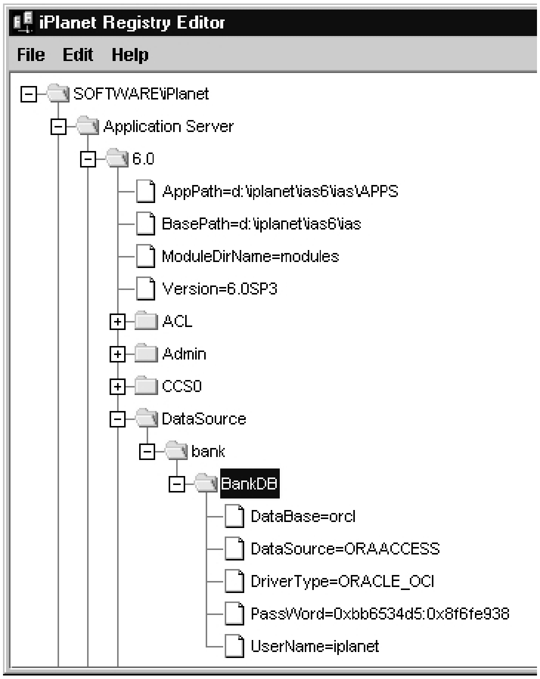
1. Start iPlanet Registry.

   (See Performing Administrative Tasks Using iASAT, for more information).

2. Open the following key:

   ```
   Software/iPlanet/Application Server/6.0/DataSource/
   ```

3. You will see a list of all the registered data sources, as shown in the following figure:



```
SOFTWARE\iPlanet
    Application Server
        6.0
            AppPath=d:\iplanet\ias6\ias\APPS
            BasePath=d:\iplanet\ias6\ias
            ModuleDirName=modules
            Version=6.0SP3
            ACL
            Admin
            CCS0
            DataSource
                bank
                    BankDB
                        DataBase=orcl
                        DataSource=ORAACCESS
                        DriverType=ORACLE_OCI
                        PassWord=0xbb6534d5:0x8f6fe938
                        UserName=iplanet
```

Select the datasource you want to delete and click `Edit>Delete`.

4. You will be prompted to confirm deletion. Click Yes to confirm. The datasource entry is deleted from iPlanet Registry.

5. Stop and start iPlanet Application Server for the changes to take effect.

NOTE You can remove a datasource that you have added, by removing the relevant entry from iPlanet Registry. You cannot remove a datasource using iASAT, as this feature is not yet implemented.

Chapter 9

Administering Transactions

This chapter describes the tasks and conceptual information necessary for administering transactions using the iPlanet Application Server Administration Tool (iASAT)

The following topics are included in this chapter:

- About the Transaction Manager
- Storing Distributed Transactions Log Data
- Administering Distributed Transactions in the Transaction Window
- Administering Distributed Transactions from the Command Line
- Setting Up Resource Managers for Distributed Transactions
- Enabling XA Error Logging
- Resolving In-Doubt Transactions
- Recovering from Log Failure

About the Transaction Manager

The transaction manager is installed with each instance of iPlanet Application Server to coordinate global transactions within a Java Server (KJS) process. Global transactions are a set of related operations that must be executed as a unit, though each operation may run in a different process.

You can use global transactions to update a database that uses one or more Enterprise Java Beans (EJBs) running concurrently for the same global transaction, from within one or more KJS processes. This occurs when an EJB triggers another EJB to run and they both participate in the same transaction. You can also update multiple databases that are distributed over different geographic locations or update multiple databases of different types (such as Oracle and Sybase).

The transaction manager runs within a KJS process and creates two files: a `restart` file and a *`restart.bak`* file. In addition, you need to provide a log file for each KJS process. You can administer these files from the command line or by using the Transaction window of iASAT.

Storing Distributed Transactions Log Data

An installation of iPlanet Application Server consists of one Administration Server (KAS) process, one Executive Server (KXS) process, and at least one Java Server (KJS) process. A transaction manager exists for each KJS.

As an iPlanet Application Server administrator, you must maintain one logical volume and its restart data for each KJS in an iPlanet Application Server installation. A logical volume is made up of one or more physical volumes. A physical volume stores the state of all ongoing transactions. If you have more than one physical volume, additional physical volumes are backups, or mirrors, of the first physical volume.

When you initially start iPlanet Application Server, it looks in the registry for the location of the directory root. In this location is an empty log file for each KJS where iPlanet Application Server will write information about the state of all ongoing distributed transactions for that process. iPlanet Application Server then creates additional files called `restart` and `restart.bak` (a backup of `restart`) for each KJS, which record the location of the log file and the state of the logical and physical volumes. Thereafter, whenever you start the server, iPlanet Application Server refers to the `restart` file for the location and state of the log file and does not refer to the registry. `Restart` and `restart.bak` are stored in the following directories:

```
iASInstallDir/ias/bin/KJS #/restart
```

```
iASInstallDir/ias/bin/KJS #/restart.bak
```

You should store `restart.bak` on a different device if possible. If `restart` becomes corrupted, iPlanet Application Server uses `restart.bak` to determine the location of the log file and state of ongoing distributed transactions. If both `restart` and `restart.bak` are corrupted, the transaction manager will become inoperable and you must “cold-start” the server. When you cold-start a server, iPlanet Application Server must look to the registry for the location of the log file as it did in its initial startup; all restart data is lost. The log file and all data will then be overwritten.

The following table lists the registry entries to which iPlanet Application Server refers along with their default values:

Table 9-1 Registry Entries

Registry Entry	Default values
`DirectoryRoot`	`iASInstallDir/CCS0/TXNMGR`
`MirrorDirectoryRoot`	`iASInstallDir/CCS0/TXNMGR_MIRROR`
`KJS #/LogVolumeDiskName`	`$DirectoryRoot/KJS #/logVol, size is 4M`

Administering Distributed Transactions in the Transaction Window

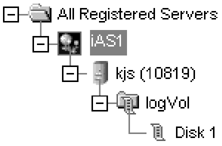
You can administer transactions using the Transaction window of iASAT.

This section describes the following topics:

- About the Transaction Window
- Configuring Transactions Per Server
- Configuring Transactions Per Process
- Configuring Resource Managers

About the Transaction Window

The left pane of the Transaction window displays a tree of nodes as shown in the following illustration:



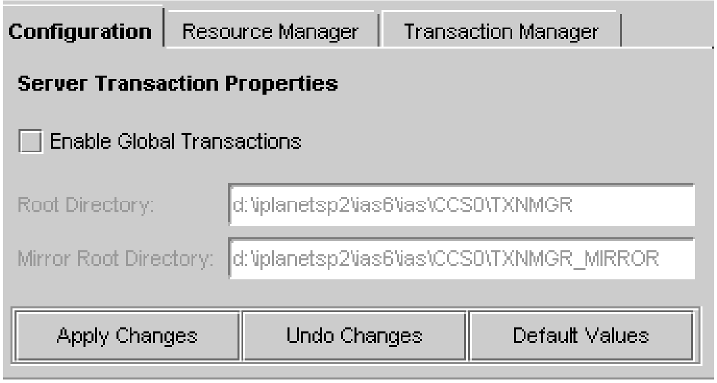
The top level of the tree lists which servers are registered with iASAT. The second level, below each registered server name, displays one or more process nodes. These nodes indicate which processes are running on each registered server. Only Java Server (KJS) processes appear in the tree because only KJS processes support transactions. The third level of the tree displays the physical volumes for each process. Finally, the fourth level of the tree displays the disks in each physical volume. See Storing Distributed Transactions Log Data for more information about physical volumes.

When you click a physical volume node, the right pane of the transactions window displays the name, status and the total size of the physical volume. You cannot edit these values.

A disk can be thought of as a partition of the physical volume. You can create an unlimited number of disks, but you cannot delete a disk once it's created. When you click a disk node, the right pane of the Transactions window displays the location and size of the selected disk.

Configuring Transactions Per Server

To change transaction settings for an application server, click a registered server in the left pane of the Transaction window. The Configuration tab appears in the right pane as shown here:



□ Enable Global Transactions

Root Directory: d:\iplanetsp2\ias6\ias\CCS0\TXNMGR

Mirror Root Directory: d:\iplanetsp2\ias6\ias\CCS0\TXNMGR_MIRROR

You can set the transaction mode. When global transactions are enabled, transactions can span across multiple heterogeneous databases and processes. When you clear the Enable Global Transactions checkbox, local transactions are enabled. Local transactions are limited to a single database/process but offer overall improved server performance over global transactions.

The selected server's current root and mirror directories are listed on the Configuration tab. Since no error checking is provided, it is not recommended that you edit these directories.

Viewing Transactions on a Selected Server

You can view transactions running on the selected server by clicking the Transaction Manager tab.

The Transactions tab displays details about all the transactions running on the selected server. For each transaction, the tab displays the following information:

- process: the Java Server process (KJS) where the transaction is running
- transaction ID: an arbitrary number used to identify the transaction
- the current state of the transaction

Click Update periodically to remove expired transactions from view and display currently running transactions in the window.

Viewing Transaction Details

To view details about a transaction, click Details.

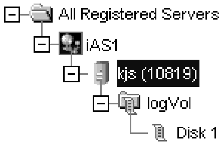
The Transactions Detail dialog box appears.

In the text box, Originator indicates where the selected transaction originates. The Participants box indicates where the transaction is currently running.

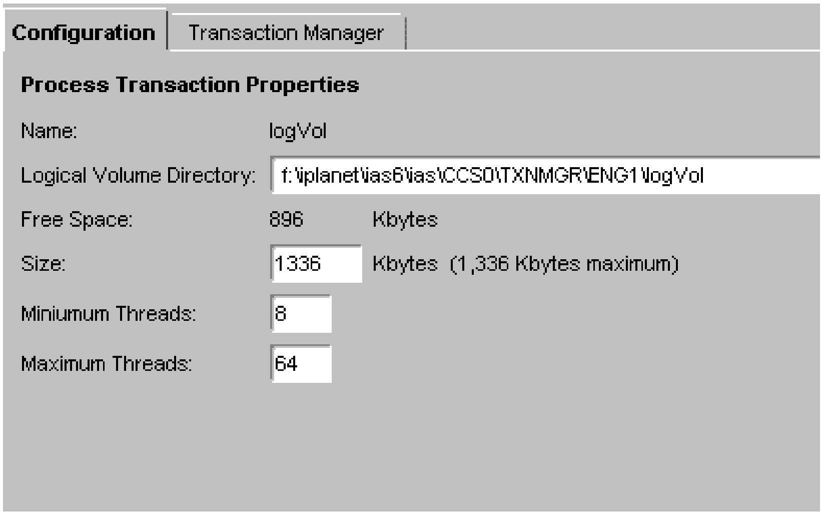
You can force the transaction into a state by clicking the appropriate button (Abort, Force Abort, Force Commit, Force Finish).

Configuring Transactions Per Process

Click the process in the left pane of the Transaction window to change transaction settings for a process on an application server.



The Configuration tab appears in the right pane as shown in the following illustration:



Process Transaction Properties

Name: logVol

Logical Volume Directory: f:\iplanet\ias6\ias\CCS0\TXNMGR\ENG1\logVol

Free Space: 896 Kbytes

Size: 1336 Kbytes (1,336 Kbytes maximum)

Miniumum Threads: 8

Maximum Threads: 64

The logical volume size for the process is displayed. You can set the size of the logical volume by entering a number in the Logical Volume Size field. A logical volume must be between 8 MB and 10 MB.

Viewing Transactions on a Selected Process

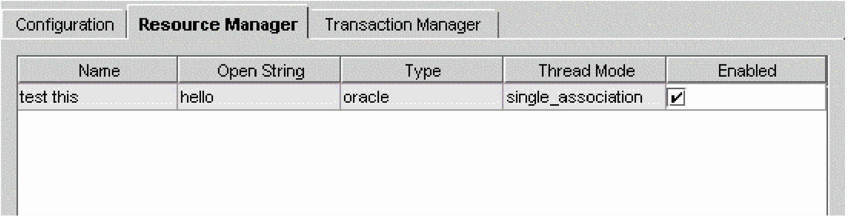
Click the Transaction Manager tab to view the details of all transactions running on the selected process. The following window appears:

The transaction ID and state appear. See Configuring Transactions Per Server for more information.

Configuring Resource Managers

A resource manager enables you connect to a database back end for global transactions. If you enable a resource manager, the transaction manager within a KJS process attempts a connection to the database when the KJS process is started.

There is one resource manager for each database the application server can access. Click the Resource Manager tab in the Transaction window to configure resource managers. The following window appears:



Name	Open String	Type	Thread Mode	Enabled
test this	hello	oracle	single_association	☑

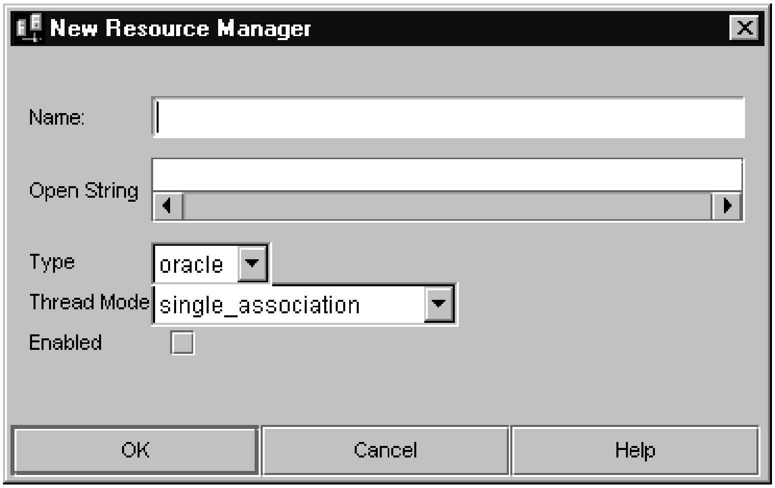
The name of each resource manager (for instance, Microsoft SQL) as well as its status (enabled or disabled) is displayed. Click the Enabled checkbox to toggle the status of each resource manager.

NOTE You must restart the server before changes to your resource manager configuration take effect.

To Add and Editing Resource Managers

To add or edit Resource Managers, perform the following steps:

1. Click Add or Modify to add or edit a resource manager. The following dialog box appears:



Name:

Open String

Type oracle

Thread Mode single_association

Enabled □

2. In the Name field, enter a value to distinguish the selected resource manager from other resource managers.

3. In the OpenString field, enter the parameters for accessing a particular database (user name, password, permissions).
4. Select the type of database from the Type drop-down box (for instance, Microsoft SQL).
5. Choose the thread mode from the drop-down box:
   - multiple_associations: the transaction manager XA (TM-XA) service performs no serialization of XA operations between threads.
   - serialize_all_operation: the TM-XA service permits a maximum of one thread to make an XA call to the resource manager client library at a time.
   - serialize_start_end: the TM-XA service ensures that only one association with the resource manager client library is attempted at a time.
   - single_association: the TM-XA service does not prevent multiple threads from attempting different associations at the same time.
6. Finally, to enable or disable the resource manager, click the Enabled checkbox.

   Only one resource manager may be enabled for each database type.

   You must restart the server before changes take effect.

Administering Distributed Transactions from the Command Line

You can also administer transactions from the command line. Invoke the command-line tool with the following script:

```
ksvradmin -l
```

The following table lists `iasadmin` commands you can execute from the command line. Once you invoke the command-line tool, each command in the following table is preceded by `iasadmin` command prompt as shown in the following example:

```
iasadmin > abort transaction
```

Table 9-2 Commands Executable from the Command Line

Command	Function	Input parameter	Output parameter
`abort transaction`	Abort a server transaction.	`DWORD tid`	

Table 9-2 Commands Executable from the Command Line *(Continued)*

Command	Function	Input parameter	Output parameter
`add trace`	Add a trace mask.	`STRING traceSpec`	
`add mirror`	Add a mirror to a logical volume.	`STRING lVol, STRING pVol, STRING diskName`	
`dump component`	Dumps the internal state of a component	`STRING componentName`	
`dump ringbuffer`	Dumps the current contents of the ringbuffer	`STRING destination`	
`expand lvol`	Expand a logical volume.	`STRING lVol, DWORD newSize`	
`expand pvol`	Expand a physical volume.	`STRING pVol, STRING diskName`	
`force transaction`	Force the outcome of a transaction.	`DWORDtid,WORD commitDesired, WORD finish`	
`help`	Display help message for given command	`{STRING commands}`	
`list trace`	Lists the current trace masks for Encina components		
`list transactions`	List unresolved transactions in the server.	`DWORD originator, DWORD participant, DWORD globalID`	`DWORD tid, WORD state` (for example, active or inactive)
`list lvols`	List all known logical volumes.	`WORD enabled`	`{STRING lVol}`
`list pvols`	List all known physical volumes.		`{STRING pVol}`
`query transaction`	Query transaction attributes.	`DWORD tid, WORD state, WORD originator,WORD participants, WORD global`	`STRING globalID, WORD state, STRING originator, {STRING participant}`

Table 9-2 Commands Executable from the Command Line *(Continued)*

Command	Function	Input parameter	Output parameter
`query logvol`	Query a log volume.	`STRING logVol`	`STRING archiveDevice, DWORD freePages, DWORD numLogFile, {STRING logFile}`
`query lvol`	Obtain information about a logical volume.	`STRING lVol`	`DWORD pageSize, DWORD size, {STRING pVol, WORD state` (e.g. clean or dirty)`, WORD isMounted}`
`query pvol`	Obtain information about a physical volume.	`STRING pVol`	`STRING lVol, DWORD chunkSize, DWORD numRegions, {STRING disk, DWORD offset, DWORD size}, DWORD totalSize`
`redirect trace`	Redirects trace to the specified destination	`STRING destination {ringbuffer, stderr, stdout, filename}`	
`remove mirror`	Remove a mirror from a logical volume	`STRING lVol, STRING pVol`	
`sync mirrors`	Synchronize mirrors of a logical volume	`STRING lVol`	
`logon`	Log on to KAS for an iPlanet Application Server installation.	`STRING name, DWORD host, DWORD port, STRING userName, STRING password, WORD autoconnect`	
`list servers`	List all the engines.		

Table 9-2 Commands Executable from the Command Line *(Continued)*

Command	Function	Input parameter	Output parameter
`set server`	Set KES as the current server and one of the engines to be the current engine. By default, the first KXS is the current server and the main engine of the KXS is the current engine.	`STRING name, WORD engNum`	
`create resourcemanager`	Create a resource manager.	`STRING name, STRING openString, STRING type, STRING threadmode, WORD isenabled`	
`delete resourcemanager`	Delete a resource manager.	`STRING name`	
`set resourcemanager`	Set an existing resource manager by modifying its open string.	`STRING name, STRING openString, STRING threadmode, WORD isenabled`	
`list resourcemanager`	List all the resource managers defined in the registry		
`get adminmode`	Return admin mode(0 or 1) for a KJS.	`WORD adminMode`	
`set directoryroot`	Sets the root directory for the transaction manager. Transaction manager events and messages are stored in this path.	`STRING <directory path>`	

Table 9-2 Commands Executable from the Command Line *(Continued)*

Command	Function	Input parameter	Output parameter
`set mirrordirectoryroot`	Sets the specified path as a backup of the transaction manager's root directory. A backup of all the transaction manager's events and messages are stored in this path. Create this path if you need a backup of transaction manager events.	`STRING <mirrordirectory path>`	
`set logvolpath`	Sets the log volume path for the transaction manager.	`STRING <log volume path>`	

Setting Up Resource Managers for Distributed Transactions

Before you can connect to resource managers to use in distributed transactions, you must perform setup tasks that are not required for local transactions.

This section contains information about the following types of resource managers:

- Oracle
- Sybase
- DB2 Unix
- Microsoft SQL Server

NOTE You must restart the server after making changes to a resource manager

Oracle

To set up an Oracle resource manager, perform the following steps:

1. Enter the open string in the following format:

   ```
   Oracle_XA+DB=<Server_Instance>+Acc=P/<user
   name>/<password>+Sqlnet=<Server Instance>+SesTm=<Session time
   out>+Threads=<Thread safe mode>
   ```

   If you are trying to connect to the bb734 instance using the user name system and the password manager, the open string appears as shown the following example:

   ```
   Oracle_XA+DB=bb734+Acc=P/system/
   manager+Sqlnet=bb734+SesTm=90+Threads=True
   ```

   Use the setting Threads=True only in the multiple_associations thread mode, which is the recommended mode for use with Oracle resource managers. Other thread modes reject this setting. Omit this parameter or use the setting Threads=False with other thread modes.

   It is strongly recommended that you use only one thread mode for all Oracle resource managers; do not mix and match thread modes for multiple resource managers.

2. Make sure the three required catalog tables for recovery exist. If they don't, create them using the following script:

   ```
   $ORACLE_HOME/rdbms80/admin/xaviews.sql (see below)

   rem

   rem $Header: xaview.sql 7020200.1 95/04/05 13:07:30 rdhoopar

   Generic<base> $ xaview2.sql Copyr (c) 1989 Oracle

   rem

   Rem
   ------------------------------------------------------------

   Rem NAME

   Rem XAVIEW.SQL

   Rem FUNCTION

   Rem Create the view necessary to do XA recovery scan of prepared

   Rem and heuristically completed transactions.

   Rem NOTES
   ```

```
   Rem The view 'XATRAN' basically combines information from two
   Rem different types of tables:
   Rem pending_trans$ & pending_sessions$
   Rem x$k2gte2
   Rem The view v$pending_xatrans$ combines and then filters
   Rem information
   Rem from the table pending_trans$ and pending_sessions$ into
   format
   Rem that satisfy XA criteria.
   Rem   Then the view v$xatrans$ combines information from x$k2gte2
   and
   Rem   v$pending_xatrans$.
   Rem MODIFIED
   Rem   cchew     07-15-92  - added fmt column
   Rem   cchew     05-22-92  - No more fmt=0 condition
   Rem   cchew     01-19-92  - Creation
   Rem
----------------------------------------------------------------
   DROP VIEW v$xatrans$;
   DROP VIEW v$pending_xatrans$;
   CREATE VIEW v$pending_xatrans$ AS
   (SELECT global_tran_fmt, global_foreign_id, branch_id
    FROM   sys.pending_trans$ tran, sys.pending_sessions$ sess
    WHERE  tran.local_tran_id = sess.local_tran_id
    AND    tran.state != 'collecting'
    AND    BITAND(TO_NUMBER(tran.session_vector),
                POWER(2, (sess.session_id - 1))) = sess.session_id)
/
   CREATE VIEW v$xatrans$ AS
   (((SELECT k2gtifmt, k2gtitid_ext, k2gtibid
   FROM x$k2gte2
    WHERE  k2gterct=k2gtdpct)
```

```
  MINUS
  SELECT global_tran_fmt, global_foreign_id, branch_id
  FROM   v$pending_xatrans$)
 UNION
  SELECT global_tran_fmt, global_foreign_id, branch_id
  FROM   v$pending_xatrans$)
/
```

NOTE For Oracle 8i, you do not need to run the XAVIEW.SQL script. Instead, connect to the database as system and run GRANT SELECT on DBA_PENDING-TRANSACTIONS to <user> (the user should be as specified in the open string).

Sybase

Sybase is only available on Solaris platforms. To set up a Sybase resource manager, perform the following steps:

1. Name the resource manager by adding entries to xa_config. The entries should be in the following format:

   ```
   [xa]
   lrm=ksample_rm
   server=ksample
   ```

2. Enter the open string in the following format:

   ```
   -U<User name> -P<Password> -N<RM name> -Txa
   ```

   For example, if you are trying to connect to ksample_rm, which is set up to connect to a ksample server instance, the open string is in the following format:

   ```
   -Uuser -Ppswd -N ksample_rm -Tevent
   ```

   If you want do not want to suppress logging user names and passwords to a trace file, use -Txa instead of -Tevent in the open string.

3. Make sure that libxa.so exists in the $SYBASE/lib directory.

   XA libraries do not come by default with Sybase client libraries.

4. Run the following scripts available in the $SYBASE/scripts/ directory:

   ```
   xacommit.sql
   ```

```
xacompot.sql
xasproc.sql
xapropt.sql
xa_ld_q1.sql
xa_ld_q2.sql
```

DB2 Unix

To set up a DB2 resource manager, perform the following steps:

1. Enter the open string in the following format:

   ```
   <DataSourceName,UserName,Password>
   ```

   For example, if you are connecting to `ksample` and using `inst1/inst1` as user name and password, the open string is in the following format:

   ```
   ksample,inst1,inst1
   ```

2. Enter the following in the DB2 configuration:

   ```
   db2 update dbm cfg using TP_MON_NAME libEncServer_nodce
   ```

   DB2 uses dynamic registration to participate in distributed transactions. On NT, DB2 needs to know which shared library implements the dynamic registration functions like `ax_reg()` and `ax_unreg()`.

3. Make sure `$DB2DIR/lib/libdb2.so` has `755` permissions.

   If it does not, the Java Server (KJS) process will crash when calling `xa_open`.

4. Make sure that `$DB2LIB/sqllib/lib/libdb2.so` has `r-x` permissions

   If it does not, the KJS process will crash upon startup.

5. Set the `CURSORHOLD` parameter to zero in the `db2cli.ini` file.

   The cursor hold feature does not work in the XA environment.

6. In the `db2cli.ini` file, set `DISABLEMULTITHREAD` to 1.

   A sample entry in `db2cli.ini` should now look like the following example:

   ```
   [ksample]
   CURSORHOLD=0
   AUTOCOMMIT=0
   ```

```
LONGDATACOMPAT=1
DISABLEMULTITHREAD=1
```

NOTE You cannot mix local and global connections using DB2 on either Solaris or Windows NT platforms. Disable all DB2 global data sources for local transactions to function properly.

Microsoft SQL Server

To set up a Microsoft SQL resource manager, perform the following steps:

1. Enter the open string in the following format:

   `Tm=`*`transaction manager's name`* `RmRecoveryGuid=GUID`

   In the iPlanet Application Server environment, `tm` is Encina.

   Find and copy the value for `RmRecoveryGuid` in the following registry entry:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\ResourceMgrID
   ```

   If this registry entry is missing, generate a GUID using the kguidgen tool.

2. Install and set up the Distributed Transaction Coordinator (DTC). You can get DTC from Microsoft's web site or from MSDN Windows NT option pack 4.0.

   When the DTC is installed, the Microsoft DTC (MS DTC) section exists in the `SOFTWARE\MICROSOFT\`hive.

   It is not necessary to install the Microsoft Transaction Server (MTS).

3. Make sure the ODBC driver on your server machine is version 3.5 or higher.

4. Make sure the following XA-related stored procedures are installed on the MS SQL Server machine where the application server connects:

   `sp_start_xact`, `sp_scan_xact`, `sp_commit_xact` or their deprecated names such as `start_xact`, `scan_xact` and `commit_xact`.

Enabling XA Error Logging

To log XA error messages, follow the directions for the type of resource manager you are using:

- Oracle
- Sybase
- DB2
- Microsoft SQL Server

Oracle

In the open string, add a log directory as shown in the following example:

```
Oracle_XA+DB=<bb734>+Acc=P/system/
manager+Sqlnet=bb734+SesTm=90+Threads=True+LogDir=/export/logs
```

where `/export/logs` is the log directory.

Make sure that the log file generated by LogDir allows administrator access only as it contains the user names and passwords for the database.

Sybase

In the open string, add a log directory as shown in the following example:

```
-Uuser -Ppswd -N ksample_rm -Tevent -L/export/logs/syb_xa_log
```

where `/export/logs` is the log directory.

Make sure that the log file generated by LogDir allows administrator access only as it contains the user names and passwords for the database.

DB2

Enter the following commands to enable the logging of XA calls and/or interfaces:

```
db2 update dbm cfg using DIAGLEVEL 4
```

```
db2 update dbm cfg using DIAGPATH $GX_ROOTDIR/logs
```

The log will be created under file name called `db2diag.log`.

XA failures appear in the following format:

```
String Title: XA Interface SQLCA  PID:28084 Node:000
```

```
SQLCODE = -998  REASON CODE: 4  SUBCODE: 4
```

Using the `REASON CODE` and `SUB CODE`, you can find the cause of an error by looking up the code in the following table:

Table 9-3 Error Codes

Code	Cause of error	Action
01 - (XAER_ASYNC)	Asynchronous operation already outstanding.	Entry is made in system log.
02 - (XAER_RMERR)	Resource manager error occurred in transaction branch.	Entry is made in system log.
03 - (XAER_NOTA)	XID is not valid.	Entry is made in system log.
04 - (XAER_INVAL)	Invalid arguments given.	Entry is made in system log. Verify content of xa open string and make necessary corrections.
04 - 01 - (xa_info)	Pointer is invalid (for example, the XAOpen string is null).	
04 - 02	Database name exceeds maximum length.	
04 - 03	User name exceeds maximum length.	
04 - 04	Password exceeds maximum length.	
04 - 05	User name specified but not a password.	
04 - 06	Password specified but not a user name.	
04 - 07	Too many parameters in the xa_info string.	
04 - 08	Multiple xa_opens generate different RM ids for the same database name.	

Table 9-3 Error Codes *(Continued)*

Code	Cause of error	Action
04 - 09	Database name not specified.	
05 - (XAER_PROTO)	Routine invoked in improper context.	Entry is made in system log.
06 - (XAER_RMFAIL)	Resource manager unavailable.	Entry is made in system log.
07 - (XAER_DUPID)	XID already exists.	Entry is made in system log.
08 - (XAER_OUTSIDE)	Resource manager doing work outside distributed transaction.	Entry is made in system log.
09	Registration (`ax_reg`) with transaction manager failed.	
09 - 01	Joining XID not found.	
09 - 02	Dynamic library specified in the `tp_mon_name` configuration parameter could not be loaded.	Ensure that the `tp_mon_name` configuration parameter contains the name of the dynamic library in the external product which has the `ax_reg()` function used for dynamic registration of transactions.
10	Attempted to start a different transaction while suspended.	
12	Unregistering (`ax_unreg`) with transaction manager failed.	
13	Ax interface failure: `ax_reg()` and `ax_unreg()` not found.	
35	Heuristic operations invalid for non-XA database.	Heuristic operation attempted against a database that only participates only as a read-only resource manager in a distributed transaction (for example, any DRDA databases like DB2 on MVS).
36	XID not known by database manager.	Invalid heuristic operation attempted on an in-doubt transaction. Wrong XID specified or a heuristic or resync operation took place since you recorded XID. Perform a Heuristic Query request to get the current list of in-doubt transactions to verify if you still need to perform your heuristic operation.

Table 9-3 Error Codes *(Continued)*

Code	Cause of error	Action
37	Transaction has already been heuristically committed.	Invalid heuristic operation attempted on an in-doubt transaction. Wrong XID specified or a heuristic or resync operation took place since you recorded XID. Perform a Heuristic Query request to get the current list of in-doubt transactions to verify if you still need to perform your heuristic operation.
38	Transaction has already been heuristically rolled back.	Invalid heuristic operation attempted on an in-doubt transaction. Wrong XID specified or a heuristic or resync operation took place since you recorded XID. Perform a Heuristic Query request to get the current list of in-doubt transactions to verify if you still need to perform your heuristic operation.
39	Transaction is not an in-doubt transaction.	XID specified is for a transaction that has ended and is waiting for the two-phase commit process to begin. Only perform heuristic operations on transactions in the two-phase commit process and have become in-doubt transactions.
40	Only rollbacks allowed for this transaction.	SQL statement attempted under a failed transaction.
69	Database log ID mismatch during DUOW re-synchronization.	Transaction manager database or resource manager database names could be referencing different database instances.
85	As a result of heuristic processing, transaction has partially committed and rolled back.	Attempting to update multiple data sources. Some data sources have been heuristically rolled back or committed, resulting in partially committed transaction that has been rolled back. To correct the data, you must manually check every data source updated by the transaction.

Microsoft SQL Server

The log file for the XA interface, `dtcxa.log`, is created under the current KJS directory.

Resolving In-Doubt Transactions

Occasionally, particularly when a Java Server (KJS) process quits suddenly, you may find “hanging” or in-doubt transactions. For Microsoft SQL Server, in order to manually commit or rollback in-doubt transactions, use DTC administrator control. This is also known as DAC. `dac.exe` is found in the `WINNT\SYSTEM32\` directory and is installed with DTC.

After starting DAC, perform the following steps to manually commit or rollback in-doubt transactions:

1. From the iASAT toolbar, click Transactions to open the Transactions window.
2. Click the Transaction Manager tab.
3. Select the transaction that you want to force and click Details.
4. Click Resolve / Abort to force rollback the transaction.

For Oracle resource managers, if you encounter a “lock held by distributed transaction” error, you must connect to the database and rollback the global transaction explicitly. To do so, perform the following steps:

1. Find out the local transaction ID that corresponds to the transaction by looking at `dba_2pc_pending`, which has all the details about pending global transactions.

   For example, type the following at the `SQLPLUS` prompt:

   ```
   SQLPLUS>select * from dba_2pc_pending
   ```

2. Rollback the transaction by typing

   ```
   rollback force transaction_id
   ```

   at the command line.

For Sybase resource managers, if you encounter a “lock held by distributed transaction” error, you must connect to the database and rollback the global transaction explicitly. To do so, perform the following steps:

1. Find out the local transaction ID that corresponds to the transaction by running `sp_xa_scan_xact`, which supplies a list of transaction identifiers.
2. Use `sp_finish_xact` with a transaction identifier and a stat (either `commit` or `rollback`) to force the branch to complete.

Recovering from Log Failure

This section describes common iPlanet Application Server log failure scenarios and explains how iPlanet Application Server can recover from these scenarios.

Logs record the state of each transaction processed by iPlanet Application Server. If this data is completely lost, some transactions - those in the prepared state before the failure - can be left in an undesirable state. You may have to resolve such transactions manually by either aborting or committing them at the resource manager. The server can then be cold-started with new volume information and the system can be brought back online. However, the transaction manager provides means for recovering from some failures without resorting to a cold-start.

These means are described in the following sections:

- Recovering from Log Disk Failure: Running Server
- Recovering from Log Disk Failure: Stopped Server
- Recovering from Loss

Recovering from Log Disk Failure: Running Server

Log volumes in the transaction manager are backed up by physical volumes. Physical volumes are backed up by disks.

A disk failure can disable a log volume which can, in turn, disable the application server. Creating a mirror of the log volume helps increase the availability of the iPlanet Application Server machine. Without a mirror, disk failure disables the iPlanet Application Server machine. If a volume is mirrored, the iPlanet Application Server machine can continue normal operation even if the log volume fails.

If one of the disks backing up the log volume fails, you can perform the following steps to restart the application server and continue normal operation:

1. Query the logical volume to obtain a list of the mirrors backing it.
2. Query the failed physical volume to obtain the size of the volume.
3. Create a disk at least as large as the physical volume.
4. Remove the old mirror.
5. Add a new mirror using the new disk.

Recovering from Log Disk Failure: Stopped Server

If a log disk fails when the server is stopped, or when the server has crashed after a disk failure, you must restart the server in administration mode.

If you know which disk has failed, perform the following steps to recover from the failure:

1. Restart the server in administration mode.
2. Remove the bad mirror.
3. Add a new mirror to replace the faulty mirror.
4. Restart the server in normal operations mode.

If you do not know which disk has crashed, restart the server in normal operations mode. The server will not start properly, but it will print the name of the failed disk.

Recovering from Loss

You can obtain information about log volume configuration from the transaction manager's `restart` file. If the `restart` file is lost, you must cold-start the server, a process that can be undesirable; when a server is cold-started, existing volume information is lost. To avoid cold-starting the server, use the backup file (`restart.bak`) that the transaction manager creates by default. Place the `restart` and `restart.bak` files on separate disks. The transaction manager can recover from the loss of one of these files, but if both files are lost, the server must be cold-started.

CAUTION Do not reuse log disks. A bug in the transaction manager prevents it from knowing whether a log disk is in use by another server. As a result, if a log disk is being used by one Java Server process (KJS1) and iASAT attempts to use the same disk as a mirror for a second Java Server (KJS2), the transaction manager destroys the contents of the disk for KJS1.

Part 3

Administering Multiple iPlanet Application Servers

Chapter 10

Configuring Multiple Servers

This chapter describes how to configure multiple iPlanet Application Server machines using iPlanet Application Server Administration Tool.

The following topics are included in this chapter:

- The Web Connector Plug-in in a Multiple-Server Enterprise
- Distributed Data Synchronization and Load Balancing
- Multicast Communication

The Web Connector Plug-in in a Multiple-Server Enterprise

The Web Connector Plug-in directs users' requests to applications on your iPlanet Application Server machine. In a multiple-server enterprise, you can specify the application server where the Web Connector Plug-in connects and logs web server requests. The application server you specify is the default server where the web connector exchanges requests and other application information. When the load balancer plug-in of iPlanet Application Server does not specify an alternate application server where application requests are forwarded, application requests are sent only to this default server.

You can also specify the application server where the Web Connector Plug-in sends the application request information for logging.

This section includes the following topics:

- Configuring the Web Connector Plug-in for Multiple Servers
- Specifying the Application Server Where Requests Are Sent

- Specifying the Application Server Responsible for Logging

Configuring the Web Connector Plug-in for Multiple Servers

When you use multiple iPlanet Application Server machines to support your enterprise application or applications, you must choose how to configure the web server to forward requests to iPlanet Application Server. These configuration options are provided by the Web Connector Plug-in. Use the configuration scenarios described in the following table to help you decide how best to configure the web connector plug-in for your enterprise:

Table 10-1 Configure the Web Connector for Multiple Servers

Configuration scenarios	What to do
One web server supporting multiple iPlanet Application Server machines without load balancing	It is assumed that the application is partitioned. Configure the web plug-in to forward requests to the application server that hosts the application objects that process the initial requests from the web browser. Use the other iPlanet Application Server machines to host the application components invoked by the objects on the first server.
Multiple web servers supporting multiple iPlanet Application Server machines without load balancing	If the application is not partitioned, configure each plug-in to forward requests to each appropriate iPlanet Application Server machine. If the application is partitioned, configure each plug-in to forward requests to an iPlanet Application Server machine that hosts the components that process the initial web browser requests. You can have multiple plug-ins connect to a single iPlanet Application Server machine.
One web server supporting multiple iPlanet Application Server machines with load balancing	The load balancing plug-in forwards application requests to the appropriate iPlanet Application Server machine. As a default, configure the web connector plug-in to forward requests to an iPlanet Application Server machine that either performs the best or hosts the application components that process the initial web browser requests.

Table 10-1 Configure the Web Connector for Multiple Servers

Configuration scenarios	What to do
Multiple web servers supporting multiple iPlanet Application Server machines with load balancing	The load balancing plug-in forwards application requests to the appropriate iPlanet Application Server machine. As a default, configure the web connector plug-ins to forward requests to each iPlanet Application Server machine, or to the iPlanet Application Server machine that either performs the best or hosts the application components that process the initial web browser requests.

When you balance application loads, the web connector plug-in works with the load balancer plug-in to automatically distribute requests across multiple iPlanet Application Server machines. This prevents all requests from going to one iPlanet Application Server machine.

If you are not balancing application loads, you must determine where a web server forwards application requests.

Specifying the Application Server Where Requests Are Sent

In a multiple application server enterprise, you can specify where the web connector sends application requests.

If you have enabled load balancing, the load balancer plug-in first dictates where the request is forwarded. However, if you have not configured the load balancer plug-in to decide where to send the request, the web connector forwards the request to the iPlanet Application Server machine you specify.

To specify the iPlanet Application Server machine to which the web server connects, perform the following steps:

1. Start iPlanet Registry Editor.

   (See About iPlanet Registry Editor)

   The editor opens and displays the keys and values that apply to the iPlanet Application Server instance.

2. Open the following key:

   ```
   SOFTWARE\iPlanet\Aplication Server\6.0\CCSO\HTTPAPI\
   ```

3. Double-click the `GXIP` String value.

   The Modify Value dialog box appears.

4. For the value data, enter the host IP address for the default iPlanet Application Server machine and click OK.

Specifying the Application Server Responsible for Logging

In a multiple-server enterprise, you can specify the application server used for web server logging.

In a single-server enterprise, the single server is the iPlanet Application Server machine where the web connector forwards application requests by default. For single-server enterprises, this value should not be changed.

In a multiple-server enterprise, the logging application server is the same server where the web connector sends application requests by default

To specify the iPlanet Application Server machine responsible for logging, perform the following steps:

1. Start iPlanet Registry Editor.

   (See About iPlanet Registry Editor)

   The editor displays the keys and values that apply to the application server.

2. Open the following key:

   ```
   SOFTWARE\iPlanet\Application Server\6.0\CCSO\HTTPLOG\
   ```

3. Double-click the `Host` String value.

   The Modify Value dialog box appears.

4. For the value data, enter the host IP address for the application server you want to perform web server logging and click OK.

5. Double-click the `Port` DWORD value.

6. For the value data, enter the port number for the Executive Server process of the same application server and click OK.

7. Close iPlanet Registry Editor.

Distributed Data Synchronization and Load Balancing

When you create a multiple application server enterprise, you must decide if you want to enable load balancing across those servers. Applications that are distributed for load balancing might have dependencies on the distributed synchronization service of the application server if those applications require state and session management.

Distributed data synchronization is configured when you install iPlanet Application Server. The installation script asks whether the server will participate in distributed data synchronization, as well as the host name and port number of the primary server. For more information about distributed data synchronization, see About Distributed Data Synchronization.

Configuring a Distributed Data Synchronization Environment

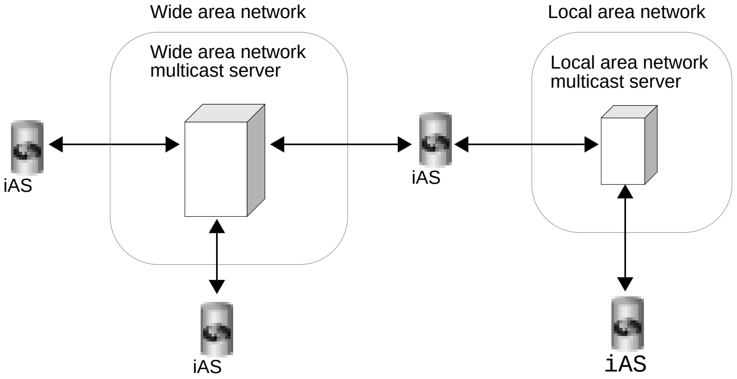
Once you install iPlanet Application Server on multiple machines, you must update the cluster keys of the servers participating in distributed data synchronization. This is done using the iPlanet Registry Editor.

Updating the keys of servers in a cluster ensures that each server has the same information about the primary server, the immediate backups, and the priority in which other servers might become a primary server in the event of a server failure.

To configure a distributed data synchronization environment, see Managing Distributed Data Synchronization.

Multicast Communication

In a multiple-server enterprise, application servers communicate with each other, for purposes of load balancing and administration, using a multicast wide area network (WAN) service. The multicast service provides a virtual server to which all messages can be posted and distributed. The application servers use an N-Way multicast configuration that allows each server to send or receive the broadcast information. The following illustration shows how this network looks:


Wide area network
Local area network
Wide area network multicast server
Local area network multicast server
iAS
iAS
iAS
iAS

Multicast services are handled by the network hardware for all servers within a local area network (LAN). For these servers, you do not have to register or change the default multicast address. When you are implementing an enterprise in a wide area network, you should use a publicly registered multicast address that allows only your iPlanet Application Server machines to communicate with each other.

How Multicast Services Apply to Load Balancing

For load balancing, you can have all servers communicate with each other, or you can create islands of servers that only balance application loads between themselves. For example, an application in New York does not need to load balance with the same application in Los Angeles. However, an application in Cupertino, Sunnyvale, and Santa Clara probably would share load responsibilities for all the users in the San Jose area.

For load balancing, multicast communication is determined by the Executive Server multicast address.

Chapter 11

Administering Multi-Server Applications

This chapter describes how to administer applications on multiple iPlanet Application Server machines using iPlanet Application Server Administration Tool (iASAT)

The following topics are included in this chapter:

- About iASAT
- Hosting Applications Locally on Multiple Servers
- Hosting Partitioned Applications on Multiple Servers
- Hosting and Deploying Applications for Load Balancing
- Changing Attributes of Distributed Application Components

About iASAT

iASAT allows you to simultaneously administer applications that are stored on multiple servers. Settings made to application components, such as Enterprise Java Beans (EJBs), distributed across multiple application servers are automatically updated across those servers. In addition, settings made to one iPlanet Application Server machine can be copied and applied to the other iPlanet Application Server machines in a group or the entire enterprise.

Using the administration tool, you can view each iPlanet Application Server machine in the enterprise and make changes to one or more servers at the same time.

To host applications on multiple iPlanet Application Server machines, you can perform either of the following tasks:

- Distribute applications or parts of applications across two or more servers to specialize request and application processing.
- Duplicate application components on two or more servers to increase application performance with load balancing.

The more servers you have to work with, the greater your choice of application hosting configurations.

The following table describes three common ways to host an application on multiple iPlanet Application Server machines:

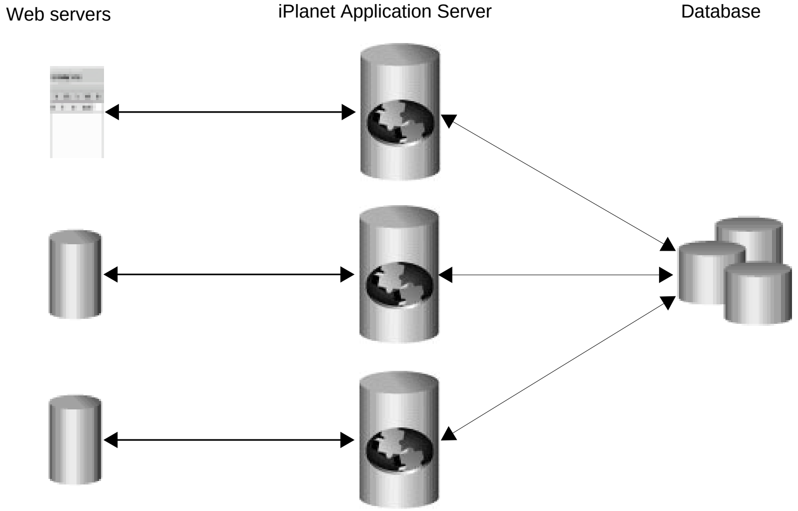
Table 11-1 Ways to Host an Application on Multiple iPlanet Application Server Machines

Hosting configuration	Description
Local	The application is installed on each iPlanet Application Server machine and uses multiple web servers to traffic requests to each server. The iPlanet Application Server machines do not communicate with each other.
Partitioned	Parts of the application are hosted on different iPlanet Application Server machines. Each server knows where the application components of the application are hosted on other servers and forwards requests to the appropriate server.
Distributed for load balancing	Parts or all of the application are duplicated on two or more iPlanet Application Server machines. You can then configure the servers to balance application-request loads.

Hosting Applications Locally on Multiple Servers

Hosting applications locally on multiple servers is the simplest of the three most common server configurations. In this configuration, you deploy the complete application on each iPlanet Application Server machine. If the application is already installed on an iPlanet Application Server machine, you can use the Deployment Tool to deploy the application to other servers.

This configuration requires that you configure each web connector plug-in to forward requests to the appropriate iPlanet Application Server machine.



Alternatively, it is possible to deploy local applications across multiple iPlanet Application Server instances while sharing a common web server and LDAP server. This configuration functions much like the first example, except that there is a single web server, and all iPlanet Application Server instances share a common configuration through the same LDAP server. This configuration has the advantage in that load balancing can be done across multiple iPlanet Application Server instances whereas the prior example requires clients to access multiple different web servers. While this scenario is possible, it may or may not be suitable for your particular application.

Hosting Partitioned Applications on Multiple Servers

To partition an application, you must divide up the application components that make up an application. Application components are then hosted by separate iPlanet Application Server machines. Partitioning applications allows you to specialize the type of processing each iPlanet Application Server machine performs.

For example, servlets responsible primarily for data access are I/O-intensive, while servlets responsible for performing calculations are CPU and active-memory intensive. To maximize your application's overall performance, you can partition the application to host these different types of servlets on separate iPlanet Application Server machines.

This section includes the following topics:

- Configuring a Partitioned Application
- Disabling and Enabling Application Components

Configuring a Partitioned Application

To configure a partitioned application, perform the following steps:

1. Deploy the complete application to all participating iPlanet Application Server machines using the iPlanet Application Server Deployment Tool.

   You can view the applications and associated modules deployed to each registered iPlanet Application Server in the left pane of the Application window. Expand a server to see the deployed applications and then expanding an application folder to see the modules in an application.

   For more information on application deployment, see the online help that is provided with the Deployment Tool.

2. Enable load balancing, which will allow each server to find application components hosted on other servers.

   For more information on load balancing, see Balancing User-Request Loads.

3. Disable specific application components on a server-by-server basis.

   See Disabling and Enabling Application Components.

While partitioning application components, if you want to view the server(s) where an application component is installed, perform the following steps:

1. Open the Application window of iASAT.
2. In the left pane of the Application window, expand the server whose application components you want to partition.
3. Open an application folder and then highlight a servlet icon for J2EE applications. For C++ applications, highlight an AppLogic icon .

Deployed application components appear in the right pane of the Application window.

a. Select an application component in the right pane of the Application window.

b. Click the Servlet Component Properties (or Application Component Properties).

   A dialog box appears displaying the application servers where the component is installed. If the selected iPlanet Application Server machine is not listed, you must deploy the `.ear` file containing the necessary application components to that machine.

c. Click OK to dismiss the dialog box.

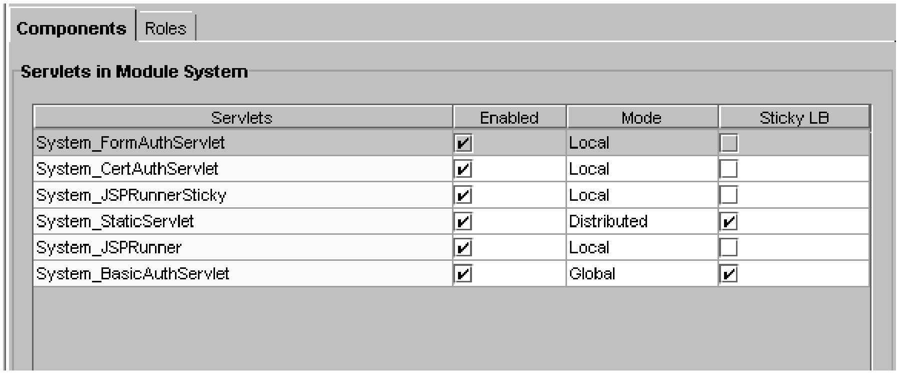
Disabling and Enabling Application Components

Disabling a component of your application (such as a servlet) stops users from accessing that component. Current requests are allowed to finish when a component is disabled, but no new requests are accepted until the component is re-enabled.

To disable an application component, perform the following steps:

1. On the iASAT toolbar, click Application to open the Application window.
2. In the left pane of the Application window, double-click the server where the application component(s) to be disabled resides.
3. Expand the folder containing the application components to disable.
4. Expand the application folder to see the application modules.
5. Select the module that contains the application component(s) you want to disable.

   The right pane of the Application window shows each application component within the module.
6. In the right pane of the Application window, locate the component to disable.
7. Locate the component(s) to disable and click the Enabled checkbox to clear the checkbox.



Servlets in Module System

Servlets	Enabled	Mode	Sticky LB
System_FormAuthServlet	☑	Local	□
System_CertAuthServlet	☑	Local	□
System_JSPRunnerSticky	☑	Local	□
System_StaticServlet	☑	Distributed	☑
System_JSPRunner	☑	Local	□
System_BasicAuthServlet	☑	Global	☑

8. Click Toggle Enabled if you want to enable or disable (toggle) all the application components in a group.

   To enable application components, click their corresponding Enabled checkboxes to select them.

9. Click Apply Changes to save your changes to your iPlanet Application Server machine.

Hosting and Deploying Applications for Load Balancing

Balancing application-request loads, or load balancing, differs from partitioning applications. Load balancing requires you to place one or more copies of an application component on multiple iPlanet Application Server machines rather than simply dividing an application's components among multiple servers (or partitioning the application). You then configure each server, allowing it to find application components on other servers.

When you deploy an application, you must decide if you want to configure the application for load balancing and, if so, how you will configure it. Choose among the following load balancing configurations:

- Balancing loads only between the servers in a production environment, if deploying to more than one iPlanet Application Server machine.

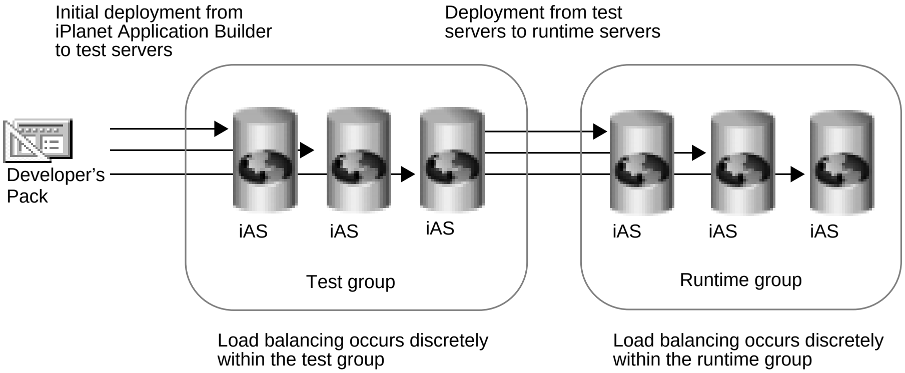
Example 1: You might have three iPlanet Application Server machines used for testing applications. Your production environment, where users' requests are actually processed, also consists of three iPlanet Application Server machines. Because the application components could be different between the two groups of servers, you do not want to enable application load balancing. Therefore, when you deploy an application from the test servers to the production servers, you should choose only to balance the loads between the destination servers.

- Balancing application loads between existing production servers and new servers that you add to the enterprise.

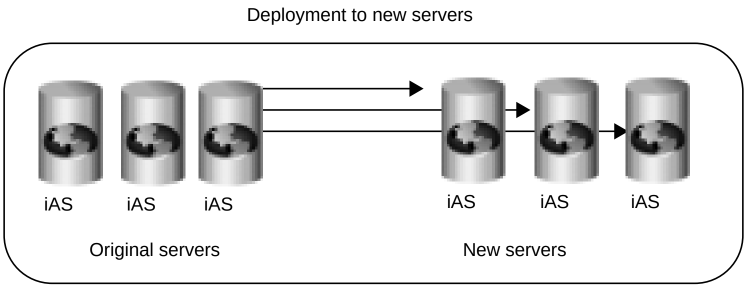
  Example 2: Suppose you scale the enterprise to include three more iPlanet Application Server machines in the production group, you can join all the servers in that group when deploying the applications from one of the existing production servers to the new servers. The application loads are then balanced between the existing servers and the new servers. (Scenario 2)

- Deploy the application locally to the server or servers (no load balancing).

The following illustration depicts a load-balancing distribution as discussed in the Example 1:


Initial deployment from iPlanet Application Builder to test servers
Deployment from test servers to runtime servers
Developer's Pack
iAS
iAS
iAS
iAS
iAS
iAS
Test group
Runtime group
Load balancing occurs discretely within the test group
Load balancing occurs discretely within the runtime group

The next illustration depicts a joining of servers when adding new servers to a group and deploying an application to those servers with the join option as discussed in Example 2.



When deploying and joining the servers, load balancing occurs between the original and the new servers

If you choose a local distribution during deployment, no application-request load balancing occurs between any of the servers.

Changing Attributes of Distributed Application Components

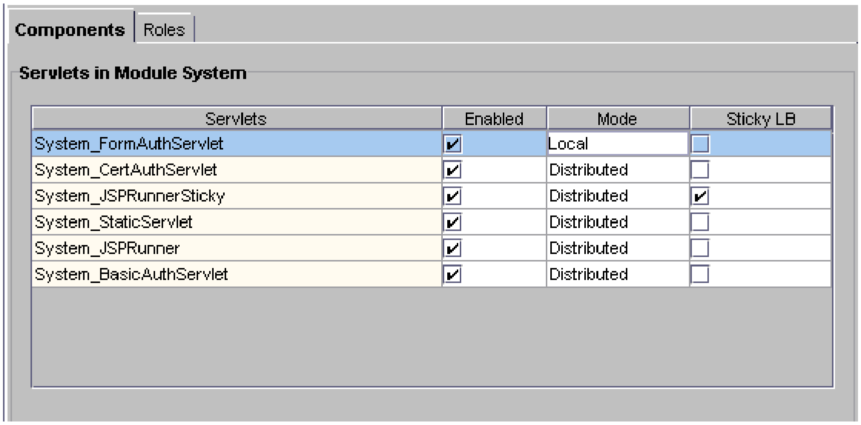
When you change such attributes as enabled sticky load balancing for an application component that is distributed across multiple servers, those changes replicate themselves on the servers where that component is hosted. Changing the distribution level (local, distributed, and global) of installed application components is useful if you previously installed an application locally, but now want to distribute the application for load balancing. You can also disable load balancing by changing a distributed application to a local configuration on the specified server.

If you change a component from a distributed or global state to a local state on one server, each server that hosts that component ceases to balance loads with the server where the distribution was set to local.

For example, an application component called ShopCart is distributed across servers A, B, and C. Should you decide to run ShopCart locally on server A, but continue to allow it to run in a distributed state across servers B and C, each server (A, B, and C) is automatically updated so that requests for ShopCart are no longer passed to server A from servers B and C. Instead, requests for ShopCart made to servers B or C are passed only between those two servers. All requests for ShopCart made to server A are processed only by server A.

To change the distribution level for an application component, perform the following steps:

1. Open the Application window of iASAT.
2. In the left pane of the Application window, expand the server for which you want to change application settings.
3. Expand the application folder and select the servlet or AppLogic icon that contains the application components you want to modify.
4. In the right pane of the Application window, select each application component for which you want to change the distribution level as follows:
   a. Local--The servlet or AppLogic runs on one iPlanet Application Server machine only
   b. Distributed--The servlet or AppLogic runs on specified iPlanet Application Server machines
   c. Global--The servlet or AppLogic runs anywhere in the enterprise.



Servlets in Module System

Servlets	Enabled	Mode	Sticky LB
System_FormAuthServlet	☑	Local	□
System_CertAuthServlet	☑	Distributed	□
System_JSPRunnerSticky	☑	Distributed	☑
System_StaticServlet	☑	Distributed	□
System_JSPRunner	☑	Distributed	□
System_BasicAuthServlet	☑	Distributed	□

5. In the Mode column, change the distribution level.
   - If you are changing the distribution level for all components in the selected group, click Toggle Mode. All application components are updated simultaneously.
   - If you are modifying the Mode from Local to Distributed or Global, you must modify the application properties to specify across which iPlanet Application Server machines load balancing is to occur.

- If you are modifying the Mode from Distributed or Global to Local, there is nothing more you need to do.

  When you change an application component's Mode to Distributed all registered servers appearing in the left pane of the Application window are added to that application component's server list. You can access the server list by clicking Application Component Properties.

6. In the left pane, under Registered Servers, choose which iPlanet Application Server machines will participate in load balancing of the selected application component. The application component must be installed on each iPlanet Application Server machine participating in load balancing.
7. If you need to register additional application servers, go to the File menu and choose New, then choose Server.
8. Repeat these steps for each application component.
9. Click Apply Changes to save your changes to the iPlanet Application Server machine.

Chapter 12

Balancing User-Request Loads

This chapter describes load balancing, which optimizes the ability of each iPlanet Application Server to process users' requests by keeping those requests balanced among several iPlanet Application Server machines.

This chapter contains the following topics:

- How Load Balancing Works
- Requirements for Load Balancing
- What Is Sticky Load Balancing?
- Selecting a Load Balancing Method
- Per Component Response Time Load Balancing
- Per Server Response Time Load Balancing
- Round Robin Load Balancing
- User-Defined Criteria Load Balancing

How Load Balancing Works

The goal of load balancing is to evenly distribute the workload between multiple iPlanet Application Server machines. When you use the iASAT to configure load balancing, you want distribute user requests as optimally as possible.

For example, if you find that many users access an application during peak usage hours, you can duplicate the application's components, such as AppLogics and servlets, on several iPlanet Application Server machines and enable load balancing. As one iPlanet Application Server machine reaches its optimal handling capacity, subsequent requests are sent to another iPlanet Application Server machine with duplicate application components. With requests evenly distributed between your servers, you can decrease response time.

You can specify the load balancing method for an iPlanet Application Server machine. The load balancing method you choose is either Web Connector Plug-in driven or iPlanet Application Server driven.

- Web Connector Plug-in driven: The Web Connector Plug-in chooses which iPlanet Application Server instance in which to send the request.
- iPlanet Application Server driven: Load balancing decisions are left to iPlanet Application Server. Server and request statistics are collected and communicated from one iPlanet Application Server machine to another in a cluster via multicasting. For more information about multicasting, see Configuring Multiple Servers.

Requirements for Load Balancing

Before your application is load balanced, the following requirements must be met:

- The application's components must be duplicated on at least two iPlanet Application Server machines or on every iPlanet Application Server machine that is to participate in load balancing.
- The distribution levels for the application components must be distributed for either specific iPlanet Application Server machines or globally to all iPlanet Application Server machines in the enterprise.

For information about enabling load balancing, see Hosting and Deploying Applications for Load Balancing.

What Is Sticky Load Balancing?

If requests within the same session are processed by more than one iPlanet Application Server machine or process, session information that is not configured to be distributed is lost. Therefore, certain application components are marked for session or “sticky” load balancing and processed on the same server, thereby eliminating the loss of session information.

When an application component is marked for sticky load balancing, it is processed by the same iPlanet Application Server machine or processed where it is initially invoked. For example, an application component called ShopCart is duplicated on two application servers for load balancing, Server A and Server B. If ShopCart is invoked by Client 1 on Server B, all subsequent sticky requests for that ShopCart from Client 1 are processed on Server B only. In other words, ShopCart “sticks” to Server B for the duration of Client 1’s session. However, at the same time, Client 2 may access ShopCart on Server A without affecting Client 1’s use of ShopCart on Server B. This maintains the integrity of state and session information for an application component that does not distribute session information.

This section describes the following topics:

- When to Use Sticky Load Balancing
- Enabling Sticky Load Balancing

When to Use Sticky Load Balancing

Sticky load balancing is necessary for application components that have interdependencies, but are running in a distributed environment. Such application components typically have the following characteristics:

- originally written to run on one machine
- depend on session information to run properly
- wrapped, not rewritten, to run in an iPlanet Application Server environment

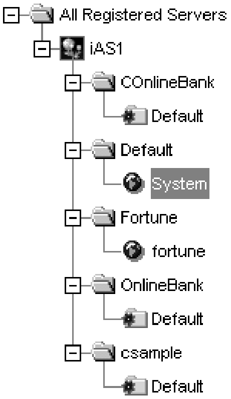
For example, a heavily used, pre-existing application is ported to run on iPlanet Application Server. Because the application is heavily used, it is distributed across several iPlanet Application Server machines to increase availability. When a user makes a request that invokes a sticky application component, the load-balancing service determines which iPlanet Application Server machine should handle that request. Once that server is chosen, all subsequent requests that use sticky application components are handled by that server. If that server becomes burdened with many users' requests, the load balancer forwards new requests to another iPlanet Application Server machine and that server processes all new session requests. This maintains an effective degree of load balancing.

Enabling Sticky Load Balancing

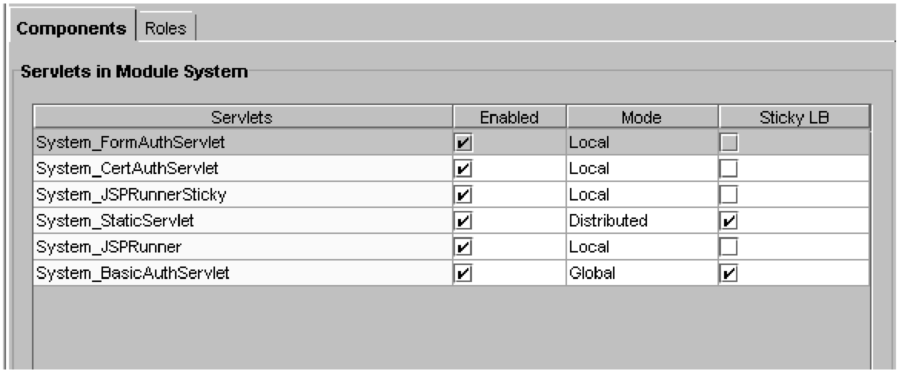
Enable sticky load balancing if there are multiple iPlanet Application Server machines and certain application components cannot distribute session and state information.

To enable sticky load balancing, perform the following steps:

1. On the iASAT toolbar, click Application to open the Application window.
2. In the left pane of the Application window, select the server where you want to enable sticky load balancing.
3. Open the application group that contains the application component or components for which you want to enable sticky load balancing.



4. In the right pane of the Application window, select the application component for which you want to enable sticky load balancing.
5. In the Sticky LB column, click the checkbox for the selected application component.



Servlets in Module System

Servlets	Enabled	Mode	Sticky LB
System_FormAuthServlet	☑	Local	□
System_CertAuthServlet	☑	Local	□
System_JSPRunnerSticky	☑	Local	□
System_StaticServlet	☑	Distributed	☑
System_JSPRunner	☑	Local	□
System_BasicAuthServlet	☑	Global	☑

Sticky load balancing is turned on for the selected component.

6. Repeat steps 4 and 5 for each application component where you want to enable sticky load balancing.

7. Click Toggle Sticky LB to select or deselect all Sticky LB checkboxes.

Selecting a Load Balancing Method

When configuring your server for load balancing, you must choose a load balancing method. Each method provides a different way to decide “who” makes the load balancing decisions. In other words, are load balancing decisions left to the server itself or does the web server plug-in make the decisions?

This section describes the following topics:

- Load Balancing with the Web Server Plug-in
- Load Balancing with iPlanet Application Server

Load Balancing with the Web Server Plug-in

If load balancing is left to the web server plug-in you can choose to load balance:

- Per Component Response Time (Default)

  The Web Connector Plug-in measures application component response time to determine where to forward an application request. This is the default load balancing choice.

- Per Server Response Time

  The Web Connector Plug-in measures server response time to determine where to forward an application request. This choice offers lower overhead than Per Component Response Time.

- Round Robin

  Requests distributed across servers based on a weighting scheme you specify

  The plug-in distributes requests across iPlanet Application Server machines according to the weights you specify. This load balancing option does not incur overhead since it is based solely on the weights that you specify and data collection regarding component or server response time is not required.

Load Balancing with iPlanet Application Server

If load balancing decisions are left to iPlanet Application Server, the application server uses a combination of hardware resource profiles (including CPU load and disk I/O) and Request Execution profiles (including result caching and servlet execution rate) to load balance individual requests. Server and request statistics are communicated from one iPlanet Application Server machine to another in a cluster via multicasting. Multicasting gives more control to the administrator, and is suitable for sophisticated scenarios.

NOTE This is the most difficult load balancing method to setup and may or may not result in increased performance. You should use this method only after trying the Web Connector Plug-in driven methods.

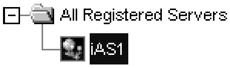
Per Component Response Time Load Balancing

Per-component response time is based on a measure of an iPlanet Application Server machine’s average response time for a specific application component.

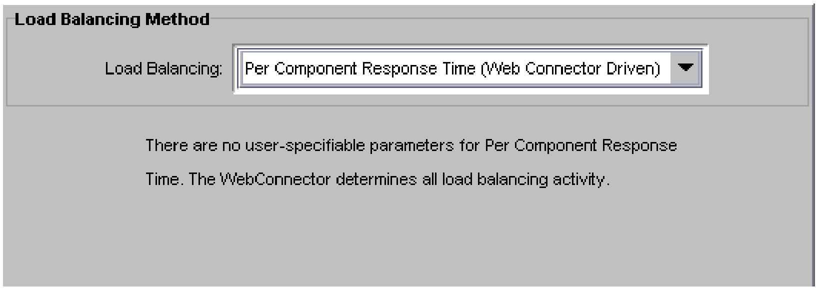
The per-component method enables richer, more detailed load balancing decisions by the Web Connector Plug-in. Keep in mind that this scenario involves a little more overhead than the per-server method. The per-component method is best suited to situations where one application component has a response time that differs widely from server to server due to varying performance characteristics.

To enable per component response time load balancing, perform the following steps:

1. On the iASAT toolbar, click Load Balancing to open the Load Balancing window.
2. In the left pane, select the server for which you want to specify the load balancing method.



3. In the Load Balancing drop-down box, choose Per Component Response Time (Web Connector Plug-in Driven) to specify the web connector plug-in will make load balancing decisions based on component response time statistics. This is the default.


Load Balancing Method

Load Balancing: Per Component Response Time (Web Connector Driven)

There are no user-specifiable parameters for Per Component Response Time. The WebConnector determines all load balancing activity.

4. Click Apply Changes to save the settings.

Per Server Response Time Load Balancing

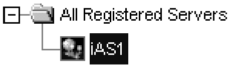
Per-server response time is based on a measure of an iPlanet Application Server machine's average response time across all the application components that machine processes.

The per-server method is best in situations where an application component has a similar response time from server to server.

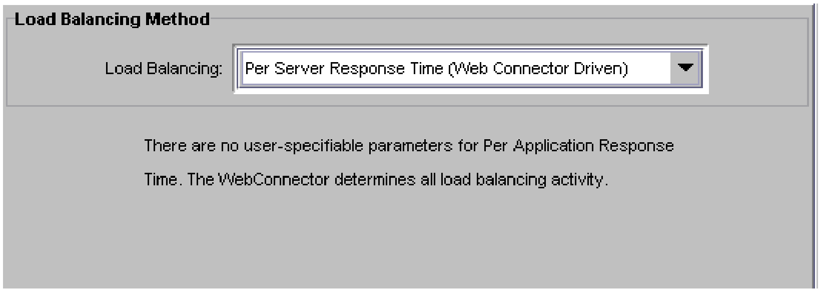
To enable per server response time load balancing, perform the following steps:

1. On the iASAT toolbar, click Load Balancing to open the Load Balancing window.

2. In the left pane, select the server for which you want to specify the load balancing method.



3. In the Load Balancing drop-down box, choose Per Server Response Time (Web Connector Plug-in Driven) to specify the Web Connector Plug-in will make load balancing decisions based on server response time statistics.



Load Balancing Method

Load Balancing: Per Server Response Time (Web Connector Driven)

There are no user-specifiable parameters for Per Application Response Time. The WebConnector determines all load balancing activity.

4. Click Apply Changes to save the settings.

Round Robin Load Balancing

When you choose round robin load balancing method, you need to specify how each iPlanet Application Server machine participating in round robin load balancing is weighted. By default, this value is one (1) unless otherwise changed. When all servers have equal weights, round robin load balancing will send equal numbers of requests to each server. You should use a weighted system when you have servers of unequal capacity. For example, if you have four machines of differing performance characteristics participating round robin, you would probably want to route more requests to the fastest machines. You do this by assigning each iPlanet Application Server machine a weight. If you assign four iPlanet Application Server machines weights of:

Machine 1 = 4

Machine 2 = 2

Machine 3 = 1

Machine 4 = 1

for every 8 requests, 4 requests will be routed to machine 1, 2 requests routed to machine 2 and so on. For a fine-grain control over the number of requests, you may want to think in terms of “how many requests out of 1000 should go to this server. For example, specifying weights of 135, 270, and 595 would offer fine-grain precision over the number of requests being sent to a server.

To setup round robin load balancing, perform the following steps.

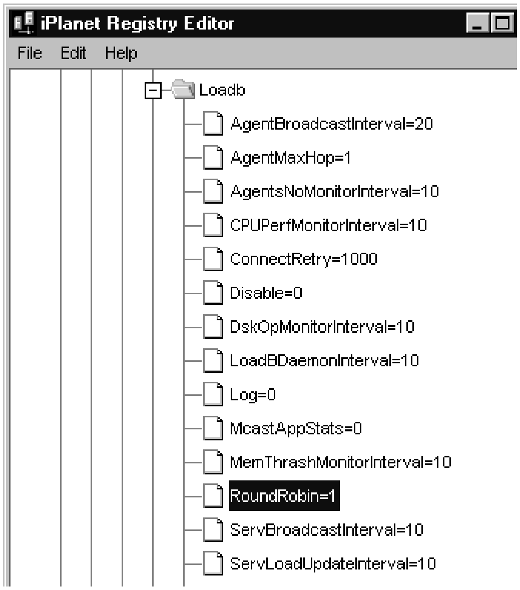
1. Start iPlanet Registry Editor.

   (See About iPlanet Registry Editor)

   The editor opens and displays the keys and values that apply to iPlanet Application Server.

2. Set the following key to 1.

   ```
   SOFTWARE\iPlanet\Application Server\6.0\CCSO\Loadb\RoundRobin
   ```



Loadb

AgentBroadcastInterval=20

AgentMaxHop=1

AgentsNoMonitorInterval=10

CPUPerfMonitorInterval=10

ConnectRetry=1000

Disable=0

DskOpMonitorInterval=10

LoadBDaemonInterval=10

Log=0

McastAppStats=0

MemThrashMonitorInterval=10

RoundRobin=1

ServBroadcastInterval=10

ServLoadUpdateInterval=10

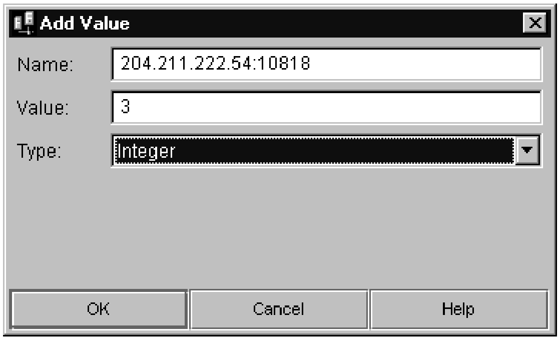
3. Highlight the following key:

   `SOFTWARE\iPlanet\Application Server\6.0\CCSO\Loadb\ServerWeights`

4. Choose Edit, then `Add Value`.

   The Add Value dialog box opens

5. Enter the name (IP Address and port number), Value (weight) for each iPlanet Application Server machines participating in round robin load balancing and set the Type to `Integer`.



Name: 204.211.222.54:10818

Value: 3

Type: Integer

For example, for three IPLANET APPLICATION SERVER machines with IP addresses (of KXS) of:

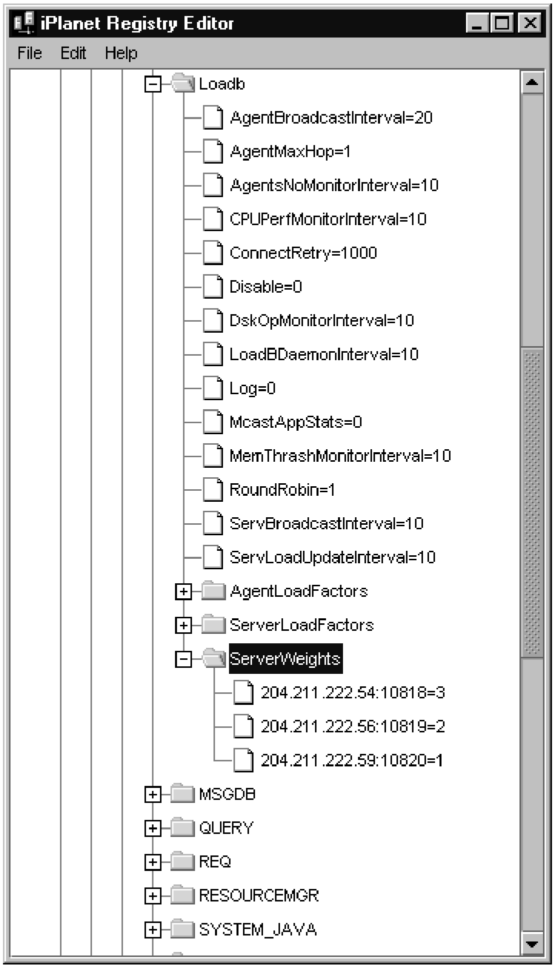
a. `204.211.222.54:10818`

b. `204.211.222.56:10819`

c. `204.211.222.59:10820`

assign the following values:

```
SOFTWARE\iPlanet\Application Server\6.0\CCSO\LoadB\ServerWeights
204.211.222.54:10818=3
204.211.222.56:10819=2
204.211.222.59:10820=1
```



```
Loadb
    AgentBroadcastInterval=20
    AgentMaxHop=1
    AgentsNoMonitorInterval=10
    CPUPerfMonitorInterval=10
    ConnectRetry=1000
    Disable=0
    DskOpMonitorInterval=10
    LoadBDaemonInterval=10
    Log=0
    McastAppStats=0
    MemThrashMonitorInterval=10
    RoundRobin=1
    ServBroadcastInterval=10
    ServLoadUpdateInterval=10
    AgentLoadFactors
    ServerLoadFactors
    ServerWeights
        204.211.222.54:10818=3
        204.211.222.56:10819=2
        204.211.222.59:10820=1
MSGDB
QUERY
REQ
RESOURCEMGR
SYSTEM_JAVA
```

Under this weighting scheme, for every 6 requests the Web Connector Plug-in will route three requests to port 10818, two requests to port 10819, and one request to port 10820.

6. Save and close the editor.

User-Defined Criteria Load Balancing

If you decide iPlanet Application Server -- not the web server plug-in --will make the load-balancing decisions for your enterprise, the load-balancing service then decides which iPlanet Application Server machine should process a request based on the weight factors you specify for the Server Load and Application Component Performance criteria. You set these factors using the iASAT's Load Balancing window. When determining weight factors, you must decide how important each criteria is for keeping your applications running optimally.

The weight factors in iASAT are initially set to default values based on the most typical applications that run on an iPlanet Application Server machine. You can adjust these factors for either Server Load criteria or Application Component criteria to optimize your specific application.

This section describes the following topics:

- Adjusting Weight Factors for Server Load Criteria
- Adjusting Weight Factors for Application Component Performance Criteria
- Adjusting Update and Broadcast Intervals
- Changing the Multicast Host Address for Load Balancing

Adjusting Weight Factors for Server Load Criteria

The Server Load value quantifies the load on an iPlanet Application Server machine while the server is processing users' requests. This value is calculated for each iPlanet Application Server machine by the load-balancing service within the respective server. You can adjust the weight factors for Server Load criteria to optimize how application requests are distributed across multiple iPlanet Application Server machines based on system resources.

The Server Load value is used as one of the criterion for calculating the Application Component Performance value. The Server Load criteria are described in the following table:

Table 12-1 Server Load Balancing Criteria

Server load criteria	Description
CPU Load	The average percentage of time all processors in a computer are in use.

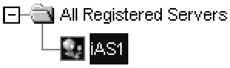
Table 12-1 Server Load Balancing Criteria

Server load criteria	Description
Disk Input/Output	The rate at which the system is issuing Read and Write operations to the hard disk drive.
Memory Thrash	The number of pages read from or written to the hard disk drive to resolve memory references to pages that were not in memory at the time of the reference.
Number of Requests Queued	The number of user and application requests a server is currently processing.
Server Response Time	Average response time from a specific server for all application components.

Each Server Load criterion is multiplied with a weight factor you set. That value is averaged with the other values to determine the final Server Load value. This value is then used as one of the Application Component Performance criteria.

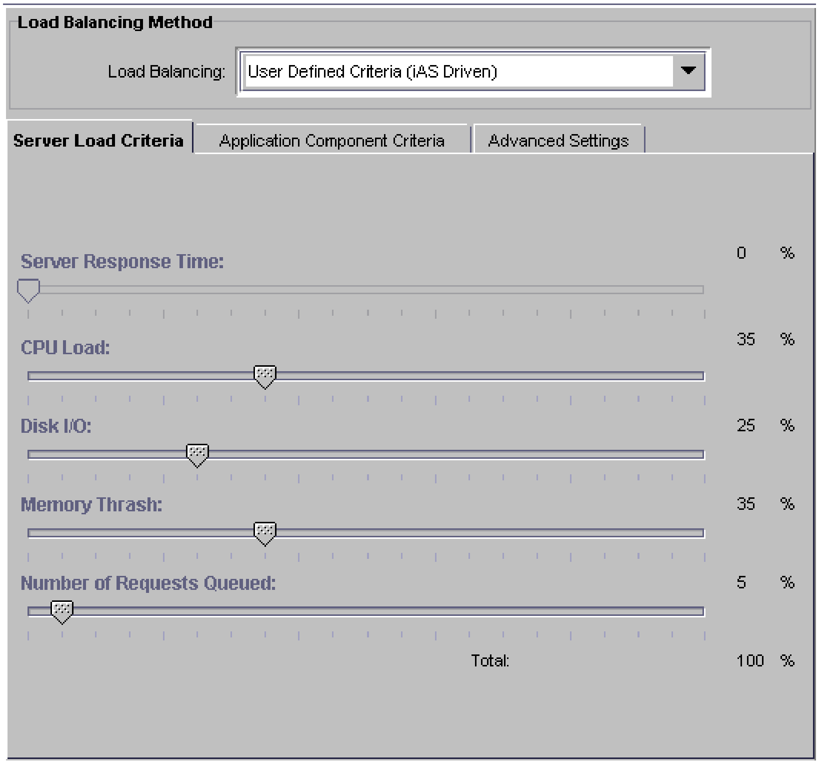
To adjust the weight factors for Server Load criteria, perform the following steps:

1. On the iASAT toolbar, click Load Balancing to open the Load Balancing window.
2. In the left pane, select the server for which you want to adjust the weight factors.



3. In the Load Balancing drop-down box, choose User Defined Criteria (iPlanet Application Server Driven) to specify the server will make load balancing decisions.

   You can then adjust the weight factors as your enterprise requires. With the Server LoadCriteria tab active, the following window appears:



Load Balancing Method

Load Balancing: User Defined Criteria (iAS Driven)

Server Response Time: 0 %

CPU Load: 35 %

Disk I/O: 25 %

Memory Thrash: 35 %

Number of Requests Queued: 5 %

Total: 100 %

4. In the right pane of the Load Balancing window, use the sliding scale markers to adjust the weight factor for each criterion. For a description of the criterion see,Adjusting Weight Factors for Server Load Criteria.

   The grand total of all weight factors must equal 100.

5. When finished, click Apply Changes to save the settings.

Adjusting Weight Factors for Application Component Performance Criteria

The Application Component Performance value represents the performance of the application components running on an iPlanet Application Server machine. This value is calculated for each application component participating in load balancing. Load balancing then occurs on an application component basis and increases distribution.

The Application Component Performance value includes five application criteria. The load-balancing service compares iPlanet Application Server machines based on the weight factor you assign for each application criterion. The server with the highest total value is chosen to process requests for that application component. The Application Component Performance criteria are described in the following table:

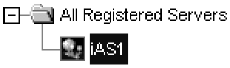
Table 12-2 Application Component Load Balancing Criteria

Application Component Performance Criteria	Description
Server Load	The value calculated for all Server Load criteria.
Cached Results Available	A flag that signals whether the results of the application component are cached. A user's request is typically processed faster when the application component's results are cached.
Lowest Average Execution Time	The time with which an application component takes to run on each iPlanet Application Server machine.
Most Recently Executed	The server that most recently ran an application component. The system on which the server is running might have cached application data, resulting in a faster execution time if that component were to be run again soon.
Fewest Executions	The number of times the application component ran on an iPlanet Application Server machine. The goal of load balancing is to equally distribute requests among all servers in the enterprise. Therefore, the server that has run the application component the least number of times is most preferred.
Application Component Response Time	Average response time from a specific server for a specific application component.

Each application criterion is multiplied by a weight factor you set. Each value is then averaged to determine the final Application Component Performance value. The final value is used by the load-balancing service to determine which iPlanet Application Server machine is best able to handle new users' requests.

To adjust the weight factors for Application Component Performance criteria, perform the following steps:

1. On the iASAT toolbar, click Load Balancing to open the Load Balancing window.
2. In the left pane, select the server for which you want to adjust the weight factors.

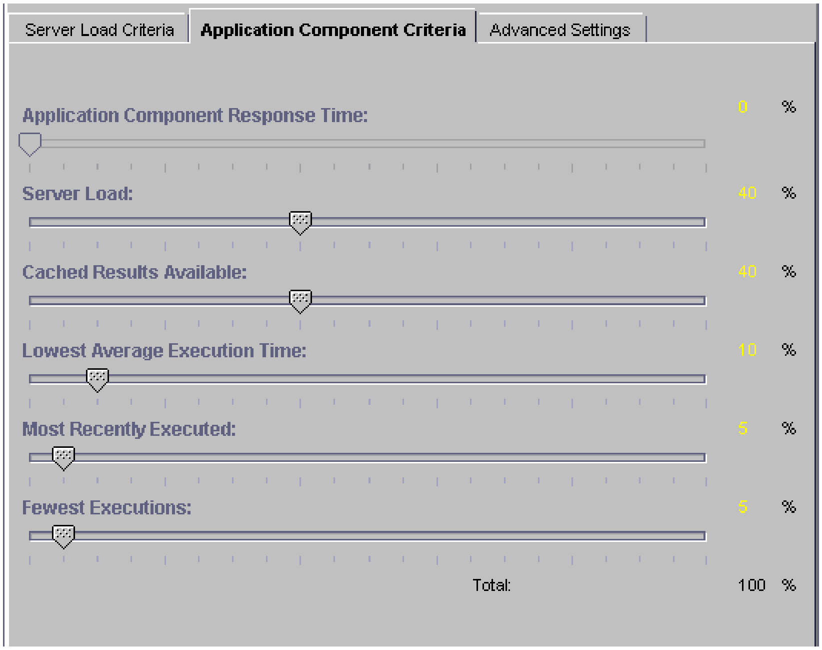


3. In the Load Balancing drop-down box, choose User Defined Criteria (iPlanet Application Server Driven) to specify the server will make load balancing decisions.

   You can then adjust the weight factors as your enterprise requires.

4. Click the Application Component Criteria tab.

   The following window appears:



Application Component Response Time: 0 %

Server Load: 40 %

Cached Results Available: 40 %

Lowest Average Execution Time: 10 %

Most Recently Executed: 5 %

Fewest Executions: 5 %

Total: 100 %

5. In the right pane of the Load Balancing window, use the sliding scale markers to adjust the weight factor for each criterion.

The grand total of all weight factors must equal 100.

6. When finished, click Apply Changes to save the settings.

Adjusting Update and Broadcast Intervals

You can set the time at which an iPlanet Application Server machine updates the Server Load and Application Component Performance criteria. If these values change frequently and drastically, it is useful to update the values often. Unfortunately, you increase the amount of work the iPlanet Application Server machine is doing by updating values frequently. You can save server resources by increasing the time between updates if the criteria values do not change often.

This theory applies to setting the broadcast intervals, as well; if values are changing often and drastically, the broadcast intervals should be short, updating servers often. This increases network traffic load, so it is important to find an optimal balance.

Broadcast and update intervals are relative to the Base Broadcast/Update Interval. This is the interval at which the load-balancing service “wakes up” and performs any updates, checks to see if any updates were received, and broadcasts any new values.

Broadcast and update intervals that are even multiples of the base interval are invoked when the load-balancing service “wakes up.” In other words, if the base value is 300 seconds, and the Server Load and Application Component Criteria broadcast intervals are at 900 seconds each, these values are broadcast every third time the load-balancing service “wakes up.” The other two times the load-balancing service awakens, it reevaluates the distribution order based on whether it received any updates from other iPlanet Application Server machines.

You can set update and broadcast intervals for several entities, as described in the following table:

Table 12-3 Broadcast Intervals

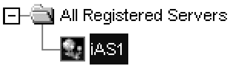
Set interval for	Description
Base Broadcast/Update Interval	The interval at which the load-balancing service “wakes up.”
Application Component Criteria	The interval at which the load-balancing service broadcasts the Application Component Performance value.

Table 12-3 Broadcast Intervals *(Continued)*

Set interval for	Description
Server Load Criteria	The interval at which the load-balancing service broadcasts the Server Load value.
Server Load	The interval at which the load-balancing service updates the Server Load value.
CPU Load	The interval at which the load-balancing service updates the CPU Load value.
Disk Input/Output	The interval at which the load-balancing service updates the Disk I/O value.
Memory Thrash	The interval at which the load-balancing service updates the Memory Thrash value.
Number of Requests Queued	The interval at which the load-balancing service updates the Number of Requests Queued value.
Max Hops	The maximum number of times a request is allowed to be passed between servers.

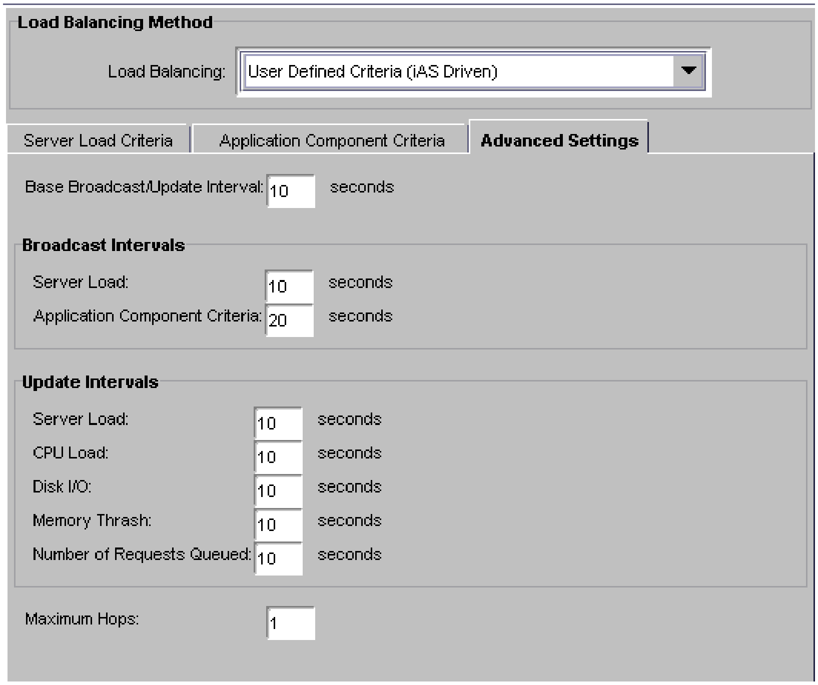
To adjust the update and broadcast intervals, perform the following steps:

1. Click the Load Balancing on the iASAT toolbar to open Load Balancing window.
2. In the left pane of the Load Balancing window, select the server for which you want to adjust the advanced settings.



3. In the Load Balancing drop-down box, choose User Defined Criteria (iPlanet Application Server Driven) to specify the server will make load balancing decisions.
4. Click the Advanced Settings tab.

   The following window appears:



Load Balancing Method

Load Balancing: User Defined Criteria (iAS Driven)

Base Broadcast/Update Interval: 10 seconds

Broadcast Intervals

Server Load: 10 seconds

Application Component Criteria: 20 seconds

Update Intervals

Server Load: 10 seconds

CPU Load: 10 seconds

Disk I/O: 10 seconds

Memory Thrash: 10 seconds

Number of Requests Queued: 10 seconds

Maximum Hops: 1

5. In the right pane of the Load Balancing window, under each interval parameter, set the time as a multiple of the base time for that parameter.
6. In the Max Hops text area, specify the maximum number of times an application component is passed between servers.
7. When finished, click Apply Changes to save your changes.

Changing the Multicast Host Address for Load Balancing

Change the multicast server host address and port number to balance application loads across networks, such as across cities. Within a network, the default address does not need to be changed unless you are experiencing a conflict.

To change the multicast host address, perform the following steps:

1. Start iPlanet Registry Editor

   (See About iPlanet Registry Editor)

2. Open the following key:

   `SOFTWARE\iPlanet\Application Server\6.0\GMS\KES`

3. Double-click the `MCastHost` String value.

   The String editor dialog box appears.

4. For the value data, specify the IP address for the new host and click OK.

5. Double-click the `MCastPort` DWORD value.

   The DWORD editor dialog box appears.

6. For the value data, specify the port number for the new host and click OK.

7. Close the editor.

   The new multicast address is in effect.

Chapter 13

Managing Distributed Data Synchronization

This chapter describes how to group iPlanet Application Server instances into clusters that participate in data synchronization.

The following subjects are described in this chapter:

- About Distributed Data Synchronization
- How Failover Keeps Data Accessible
- What Is a Cluster?
- How a Cluster Communicates
- Configuring Clusters
- How Sync Server Prioritization Improves Coordination

About Distributed Data Synchronization

Distributed data synchronization maintains the integrity of shared state and session information across multiple iPlanet Application Server instances. This is crucial for partitioned and distributed applications that are hosted on multiple iPlanet Application Server instances.

In most enterprises, several iPlanet Application Server instances support one or more distributed applications. For such distributed applications to run successfully, each server must have access to the relevant information for that application, such as state and session information.

Support for this distribution of information is provided through a system-level distributed data synchronization service that is built into iPlanet Application Server.

How Failover Keeps Data Accessible

The distributed data synchronizer is a system-level service that controls how distributed data, such as application session information, is maintained and made accessible across multiple iPlanet Application Server instances.

Each iPlanet Application Server instance is made up of the following four "engines:"

- Administrative Server (KAS) – The Administrative Server brings up and monitors the other engines and makes sure that any engines that fail are brought up again.
- Executive Server (KXS) – Only an Executive Server can be the primary synchronization engine (the synchronizer) for an iPlanet Application Server cluster.

  In a cluster of iPlanet Application Server instances, one of the Executive Servers maintains the distributed (synchronized) information and sets up server roles for all the other servers participating in the cluster. All engines in a cluster know how to access this primary engine and the information that is on this primary engine.
- Zero or more Java Servers (KJS)
- Zero or more C++ Servers (KCS)

  If KJS or KCS fails, KAS simply restarts the failed engine. However, if KXS fails, KAS performs the following actions:

  - Brings KXS back up in the currently appropriate role. This role is determined in synchronization with KXS engines in the cluster, and may not necessarily be the role previously occupied.
  - Brings down the KJS and KCS engines.
  - Brings the KJS and KCS engines back up.

What Is a Cluster?

A cluster is a group of iPlanet Application Server instances that share information related to the state of individual user sessions. The primary benefit of using clusters is to improve reliability, to continue user sessions even after process and hardware failures. The secondary benefit is improved load distribution across all available resources.

Servers in a cluster can belong to the same network or to different networks, or to different subnets within the same network. While keeping the servers in a cluster on a common subnet enhances performance, this is not necessary. Members in a cluster communicate through TCP/IP, which uses the IP Addresses and port numbers of the servers in the cluster, to access the KAS and the KXS engines of each instance. As long as this communication takes place, the cluster will be operable. See Using Firewalls for Security, for more information on how this communication takes place.

In a cluster, the state/session data is stored in the memory of the KXS process for the Primary Sync Server. The configuration data of the cluster is stored in Directory Server. All the iPlanet Application Server instances in your cluster can share a single Directory Server; if the iPlanet Application Server instances in your cluster do not share a single Directory Server, cluster settings must be copied from one Directory Server to another so each server has access to identical cluster information.

This section describes the following topics:

- Setting Up Data Synchronization
- Server Roles Within Clusters

Setting Up Data Synchronization

To set up data synchronization between servers, you need to decide the role that you want a server to perform in the cluster. You can then edit each cluster entry to set up the server roles and register the cluster with the synchronizer service. Finally, you need to start each iPlanet Application Server in the order that is determined by server roles.

Server Roles Within Clusters

You can configure a server's role within a cluster as either a Sync Alternate or a Sync Local. The server that you start first becomes your Primary Sync Server and the server that you start next becomes your Backup Sync Server. This goes on till the number of backup servers you have defined have been started. If you have defined 2 backup servers, then the 2 servers that you start after the Primary Sync Server is started play the role of backup Sync Servers.

The next server that is started (once the defined number of backup servers have been started), is your Sync Alternate. This server takes over if the Primary and the backups fail.

See Example: Coordination Within a Seven-Server Cluster, to learn more about how servers in a cluster coordinate with each other, while performing the roles that they have been assigned.

The roles that the servers in a cluster can perform are described in the following table.

Table 13-1 Roles for Data Synchronization in a Cluster

Server role	Description
Sync Server	Any iPlanet Application Server instance can be identified as Sync Primary. The Sync Server category contains the Sync Primary, Sync Backups and Sync Alternates. All Sync Servers are listed in the `SyncServers` key of iPlanet Registry.
Sync Primary	The server that is the primary data store, with which all other cluster members communicate for the latest distributed data information. The first iPlanet Application Server to be started in a cluster must be a Sync Server, and that Sync Server becomes the Sync Primary for the cluster simply because it is started first.
Sync Backup	Any number of Sync Servers, up to a maximum number (`MaxBackups`) set by you, that mirrors the information on the Sync Primary. Because each Sync Backup increases the load on the cluster, weigh safety against performance impacts when deciding how many backups to assign. If the Sync Primary becomes inaccessible, the Sync Backup with the highest priority (which is the lowest integer value) relative to other Sync Backups becomes the next Sync Primary.

Table 13-1 Roles for Data Synchronization in a Cluster

Server role	Description
Sync Alternate	A server listed in the SyncServers key in iPlanet Registry that is eligible to become a Sync Backup. If the number of Sync Backups falls below the set maximum, the Sync Alternate with the highest priority relative to other Sync Alternates is promoted to Sync Backup. Each Sync Alternate performs work similar to that of a Sync Local until the Sync Alternate is promoted to Sync Backup.
Sync Local	A server that uses data synchronization services, but is not eligible to become a Sync Primary, Sync Backup, or Sync Alternate. Sync Locals can use, create, and destroy all distributed data, but are never responsible for maintaining that data. Sync Locals are not listed in the SyncServers key in iPlanet Registry. However, the SyncServers list in every registry in the cluster contains identification and priority information for each of the Sync Servers in the cluster. Each Sync Local contacts each of the servers listed in its SyncServers key in iPlanet Registry until the Sync Local finds the Sync Primary, at which time the Sync Local becomes active in the cluster. If the Sync Local goes through its entire SyncServers key in iPlanet Registry without finding the Sync Primary, the Sync Local assumes that the cluster is down, and acts as a local server. Sync Locals communicate only with the Sync Primary, and the other servers in the cluster are not aware of them.

How a Cluster Communicates

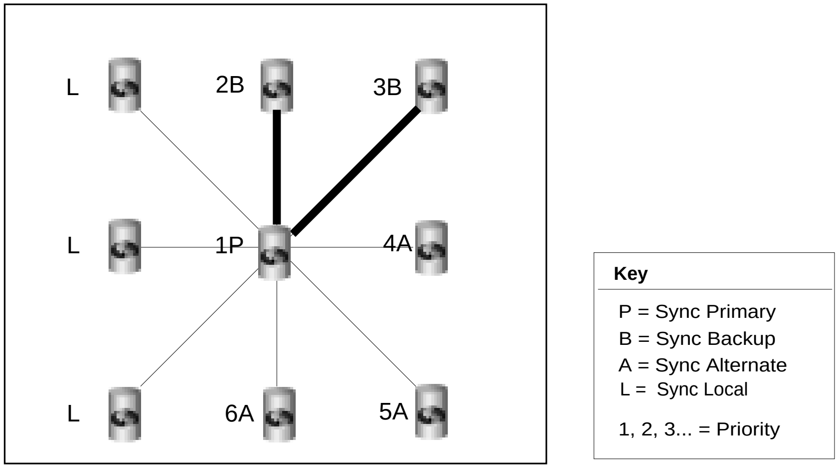
Servers in a cluster need to communicate with each other. To enable this communication, it is necessary that each server identifies with the cluster to which it belongs. An iPlanet Application Server instance becomes an active part of a cluster when you map its synchronizer to the cluster. This procedure is described in Mapping the Synchronizer to the Cluster.

When an application component requests “write” access to a distributed data source, the write occurs first on the Sync Primary. When the data changes on the Sync Primary, the Sync Primary immediately updates the Sync Backups.

You can map the synchronizer for each iPlanet Application Server instance to only one cluster at a time.

Information Flow Within a Cluster

Sync Backups, Sync Alternates, and Sync Locals communicate with the Sync Primary in a star configuration, as shown in the following illustration:



In this illustration, notice that all servers are communicating with the Sync Primary. Also note that no Sync Local is assigned a priority number.

Note also that this illustration is an ideal representation of a cluster that has probably just started and has not experienced failover, as the priority numbers correspond gracefully to the currently assigned roles.

Configuring Clusters

You can set up and manage clusters using the iPlanet Application Server Administration Tool (iASAT). iASAT is a stand-alone graphical user interface tool, using which you can manage all the administrative aspects of iPlanet Application Server. When you configure clusters using iASAT, the relevant back-end entries are created in iPlanet Registry, which you can verify and edit.

You can also configure clusters directly in iPlanet Registry, using iPlanet Registry Editor. However, it is recommended that iASAT be used to create and manage clusters.

The following sections describe how to create and manage clusters, using both iASAT and iPlanet Registry Editor.

- Configuring Clusters Using iASAT
- Configuring Clusters Using iPlanet Registry Editor
- Determining Sync Server Priority
- Setting Cluster Parameters
- Mapping the Synchronizer to the Cluster

Configuring Clusters Using iASAT

When you install iPlanet Application Server on your instance, your host server is automatically taken as the default cluster, with a single node. You can add other registered iPlanet Application Server instances to the existing cluster (your host server), or create a new cluster, or create a completely new cluster.

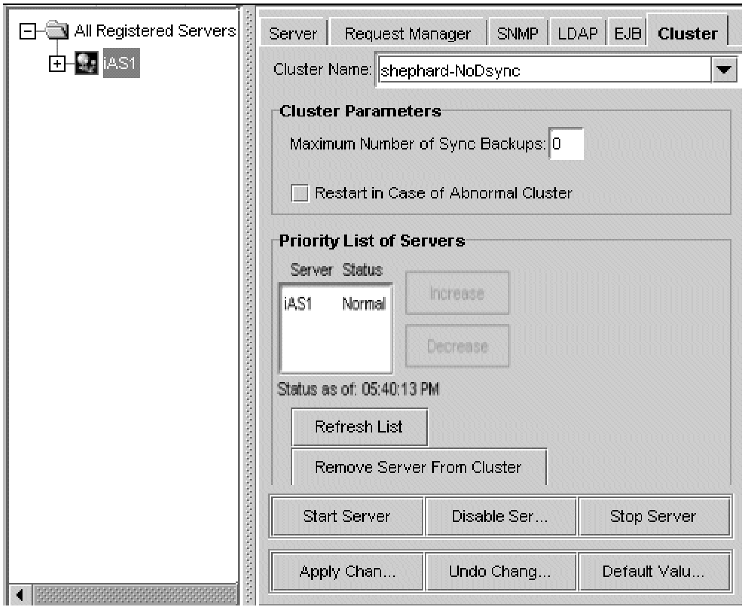
This section describes the following topics:

- To Create a New Cluster
- To Add a Server to a Cluster
- To Remove a Server From a Cluster

To Create a New Cluster

To create a new cluster using iASAT, perform the following tasks:

1. In the iASAT toolbar, click General to open the General window.
2. From the left pane of the General window, select the iPlanet Application Server instance with which you want to create a new cluster.
3. In the right pane of the General window, click the Cluster tab, as shown in the following figure:



Cluster Name: shephard-NoDsync

Cluster Parameters

Maximum Number of Sync Backups: 0

□ Restart in Case of Abnormal Cluster

Priority List of Servers

Server	Status
iAS1	Normal

Status as of: 05:40:13 PM

4. In the Cluster Name drop-down list, delete the default cluster name entry, using either the Backspace or the Delete key.

5. Provide the name of the new cluster and press Enter. You can choose any unique name for the new cluster.

6. Click Apply Changes. You need to start the server again for the new cluster configuration to take effect.

When you restart the server, you can add other iPlanet Application Server instances to the new cluster as described in the following section.

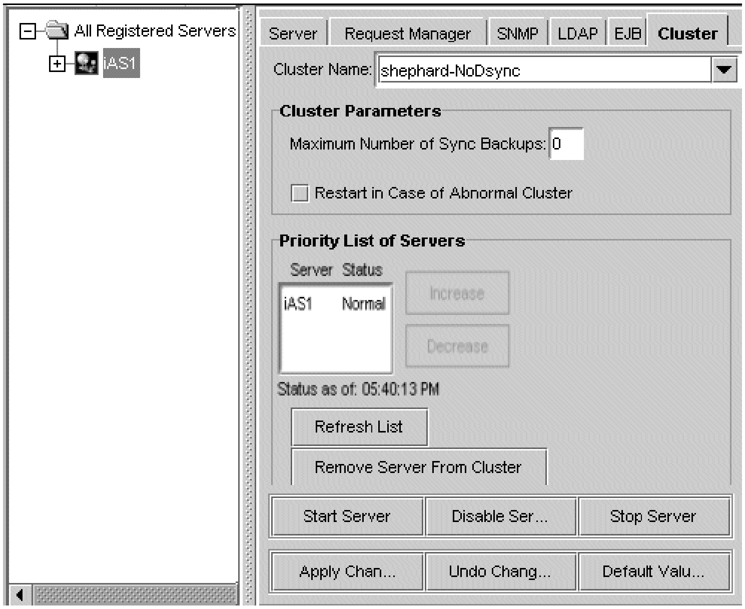
To Add a Server to a Cluster

To add an unassigned server to a cluster or to reassign a server to a different cluster using iASAT, perform the following steps:

1. From the iASAT toolbar, click General to open the General window.

2. Click the Cluster tab.

   The following window appears:



Cluster Name: shephard-NoDsync

Cluster Parameters

Maximum Number of Sync Backups: 0

□ Restart in Case of Abnormal Cluster

Priority List of Servers

Server	Status
iAS1	Normal

Status as of: 05:40:13 PM

A list of all registered servers is displayed in the left pane of the General window. A list of the server(s) belonging to the existing cluster(s) is displayed in the Priority List of Servers box, in the right pane of the General window. These servers are sorted by their priority in a cluster.

The Priority List of Servers box also shows the cluster status of a server. Server conditions can be Normal, Dual Primary or No Primary. Click the Refresh List button to immediately update the Priority List of Servers box. By default, this box is updated every 15 seconds.

3. In the left pane of the General window, click the name of the server you want to add to a cluster.

4. From the Cluster Name drop-down box, select the name of the cluster to which you want to add the server.

   The Cluster Name drop-down list is populated with the cluster names to which all *registered* servers belong.

5. Click Apply Changes.

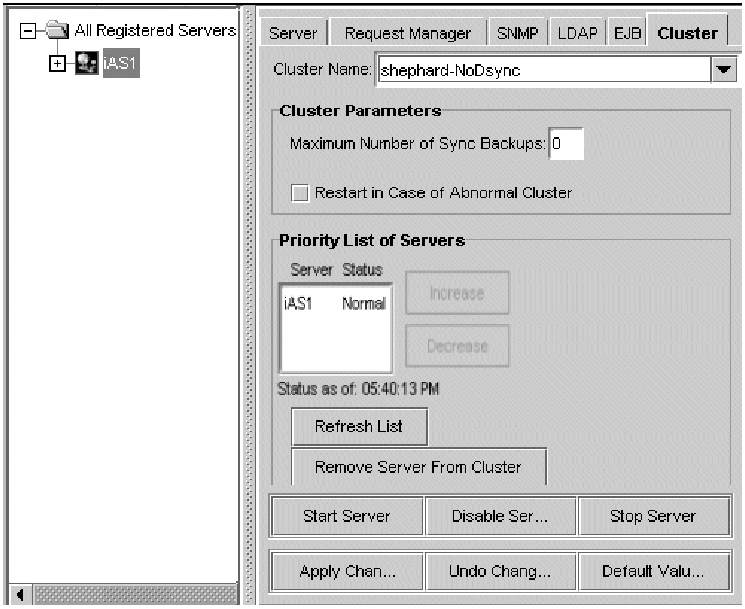
6. Stop and start every server in the cluster, including the server you just added.

NOTE If you remove a server from a cluster and add it to another, you need to restart all the servers in both clusters, for the change in configuration to take effect.

To Remove a Server From a Cluster

To remove a server from a cluster, perform the following steps:

1. From the iASAT toolbar, click General to open the General window.
2. From the left pane of the General window, select an iPlanet Application Server that is a member of the cluster from which you want to remove a server.
3. Click the Clusters tab in the right pane of the General window, as shown in the following figure:



Cluster Name: shephard-NoDsync

Cluster Parameters

Maximum Number of Sync Backups: 0

☐ Restart in Case of Abnormal Cluster

Priority List of Servers

Server	Status
iAS1	Normal

Status as of: 05:40:13 PM

A list of all registered servers is displayed in the left pane of the General window. A list of the server(s) belonging to the existing cluster(s) is displayed in the Priority List of Servers box, in the right pane of the General window.

4. In the Cluster Name drop-down list, select the cluster from which you want to remove a server. The Priority List of Servers box shows all the servers belonging to the selected cluster.
5. In the Priority List of Servers box, select the server you want to delete and click Remove from Cluster.
6. Click Apply Changes.
7. Shut down and restart every server in the cluster, including the server you just removed.

NOTE

- A server that is not a member of a cluster, hence not participating in data synchronization, will be listed under `<hostname>-NoDsync`, in the cluster list. You cannot remove a server from the `<hostname>-NoDsync` list.
- If you want to rename a cluster, delete the cluster name in the Clusters tab, in the General window of iASAT. Type a new name for the cluster and click Apply Changes.

Configuring Clusters Using iPlanet Registry Editor

When you configure clusters using iASAT, the necessary back-end entries are created in iPlanet Registry. You can directly create these entries in iPlanet Registry and configure clusters, without using iASAT. Although this facility is available, it is strongly recommended that you use iASAT to set up and manage your clusters.

This section describes the following topics:

- To Create a New Cluster
- To Add a Server to a Cluster
- To Remove a Server From a Cluster

To Create a New Cluster

When you install iPlanet Application Server, a default cluster is automatically created, using the host server. The easiest way to set up and configure clusters is to modify the back-end entries created in iPlanet Registry for this default cluster.

The default cluster bears the name <hostname>-NoDsync, where hostname is the name of your local instance. For instance, if you install iPlanet Application Server on a machine named *Bozo*, the default cluster entry will be named Bozo-NoDysnc. The default values for this cluster are configured during iPlanet Application Server installation. You can configure the existing default cluster, or configure a completely new cluster, as described in the sections that follow.

To create a new cluster, perform the following tasks:

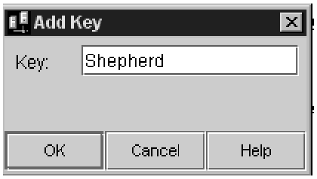
1. Start iPlanet Registry. (See About iPlanet Registry Editor)

NOTE Note that you can have multiple instances of iPlanet Application Server on UNIX systems, each with its own iPlanet Registry. Ensure that you edit the iPlanet Registry that belongs to the server instance for you want to map the synchronizer.

2. Open the following key:

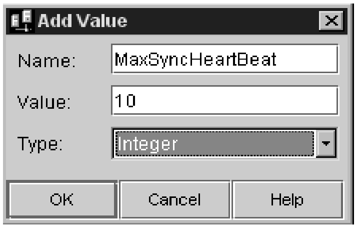
   SOFTWARE\iPlanet\Application Server\Clusters\

3. From the Edit menu, choose Add Key. The Add Key dialog box appears, as shown in the following figure:



Key: Shepherd

4. In the Key field, provide a name for the new cluster and press OK. A new cluster is created. You now need to configure the cluster, as described in the ensuing steps.
5. Select the new cluster that you created, and choose Add Value from the Edit menu. The Add Value dialog box appears, as shown in the following figure:



Name: MaxSyncHeartBeat

Value: 10

Type: Integer

When you add the required values to your cluster, make sure that the type for each value is set to Integer. You can do this by selecting Integer from the Type drop-down list, in the Add Value dialog box.

When you finish adding a value, click OK to confirm. To add another value, select the new cluster you have created and choose Add Value from the Edit menu.

6. You need to add the following values to the new cluster:

   ❍ `MaxBackups`

     This value indicates the maximum number of servers that can be assigned as backup servers. The default value for this key is 1.

NOTE A key named `MaxHops` is automatically configured below the `MaxBackups` key when you create a new cluster using iASAT. This key relates to an unsupported feature. If you create clusters using iPlanet Registry Editor, you need not create an entry for this key.

   ❍ `MaxSyncHeartBeat`

     This value specifies the maximum number of heartbeat messages that an engine will send to any other engine. The heart-beat mechanism is used to detect an abnormal cluster condition. The default value for this key is 10.

   ❍ `SyncHeartBeatInterval`

     This value specifies the number of seconds between two heartbeat messages sent from one server to another. The default value is 30.

- `SyncTimerInterval`

  This key specifies the intervals, in seconds, at which the synchronization service wakes up and checks to see whether any data has expired. This key specifies how often the timer thread goes through the node list and removes all the nodes that have expired. The default value is 30.

  See Setting Cluster Parameters, for more information on configuring these values for best results.

7. When you finish adding these values, select the new cluster key and choose `Add Key` from the Edit menu.

8. In the AddKey dialog box that appears, type `SyncServers`.

   This creates a folder called SyncServers under the new cluster key that you have created. When you add servers to your cluster, you will need to add them under this key.

   Your new cluster is created and configured. You can now add servers to your cluster, as described in the next section.

Adding a Server to a Cluster

To add servers to a cluster, perform the following tasks:

1. Start iPlanet Registry.

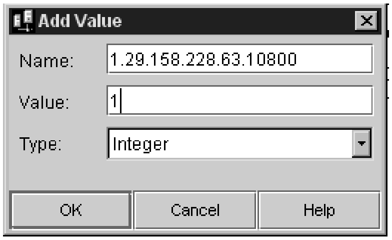
   (See About iPlanet Registry Editor)

NOTE	Note that you can have multiple instances of iPlanet Application Server on UNIX systems, each with its own iPlanet Registry. Ensure that you edit the iPlanet Registry that belongs to the server instance for which you want to map the synchronizer.

2. Open the following key:

   ```
   SOFTWARE\iPlanet\Application
   Server\Clusters\<clustername>\SyncServers
   ```

3. From the Edit menu, choose `Add Value`. The Add Value dialog box appears, as shown in the following figure:



Add Value

Name: 1.29.158.228.63.10800

Value: 1

Type: Integer

4. In the Name text field, provide the IP Address of the instance which you want to add to the cluster. Type a colon (:) after the IP Address and enter the KXS port number, as shown in the following example:

   `host IP address:KXS port number`

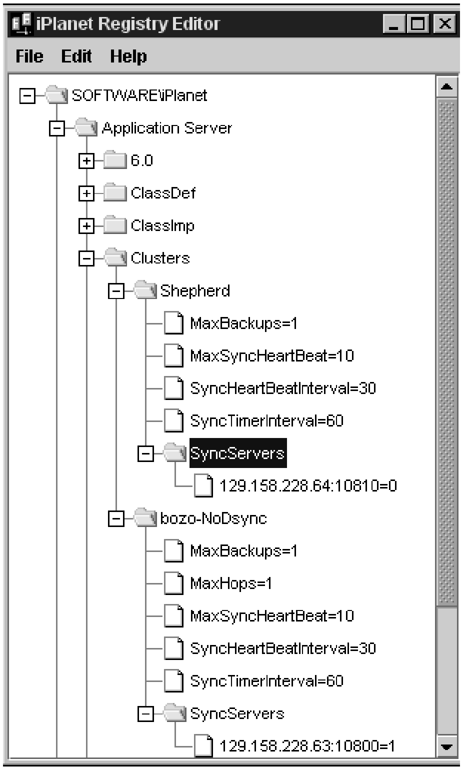
   Your entry should look like this.

   `129.158.228.63:10800`

5. In the Value text field, provide a value that indicates server priority in the cluster. For example, `0` indicates the highest priority and `1` indicates the next level of priority. The lowest priority value is `65,535`.

6. Select Integer from the Type drop-down list and click OK to register your entry in iPlanet Registry.

   When you finish adding the IP Address of a server to the `SyncServers` key, the back-end entry in iPlanet Registry must look like the following example:



```
SOFTWARE\iPlanet
    Application Server
        6.0
        ClassDef
        ClassImp
        Clusters
            Shepherd
                MaxBackups=1
                MaxSyncHeartBeat=10
                SyncHeartBeatInterval=30
                SyncTimerInterval=60
                SyncServers
                    129.158.228.64:10810=0
            bozo-NoDsync
                MaxBackups=1
                MaxHops=1
                MaxSyncHeartBeat=10
                SyncHeartBeatInterval=30
                SyncTimerInterval=60
                SyncServers
                    129.158.228.63:10800=1
```

You can now continue to add more servers to your cluster. When you finish adding servers to your cluster, you need to map the servers to the cluster to which they belong. See Mapping the Synchronizer to the Cluster.

To Remove a Server From a Cluster

You can remove a server from a cluster that you have added, by deleting the server's IP Address entry from the `SyncServers` key.

To remove a server from a cluster using iPlanet Registry Editor, perform the following tasks:

1. Start iPlanet Registry.

   (See About iPlanet Registry Editor)

NOTE Note that you can have multiple instances of iPlanet Application Server on Solaris systems, each with its own iPlanet Registry. Ensure that you edit the iPlanet Registry that belongs to the server instance for you want to map the synchronizer.

2. Open the following key:

   ```
   SOFTWARE\iPlanet\Application
   Server\Clusters\<clustername>\SyncServers
   ```

   You will see the IP Address and port number entries of the servers that have been added to the cluster.

3. To remove a server, select the required IP Address entry and choose Delete from the Edit menu. Alternatively, you can select the required entry and press the Delete key.

4. You will be prompted to confirm the action. Click Yes to confirm. The server is removed from the cluster.

NOTE

- Note that you can add the server that you removed, again to the same cluster, or to a different cluster. See To Add a Server to a Cluster for more information.
- You can’t rename a cluster in iPlanet Registry. You will have to delete the cluster name entry and create a new cluster configuration. See To Create a New Cluster for more information.

Determining Sync Server Priority

You can set priority values to sync servers in a cluster. Priority values are used to select between Sync Servers in the same status (either between a group of Sync Backups or between a group of Sync Alternates). Only the order in which instances of iPlanet Application Server are started, not priority, determines which server should be the Sync Primary and which Sync Servers will start out as Sync Backups or Sync Alternates.

You can set the sync server priority using either iASAT or iPlanet Registry Editor.

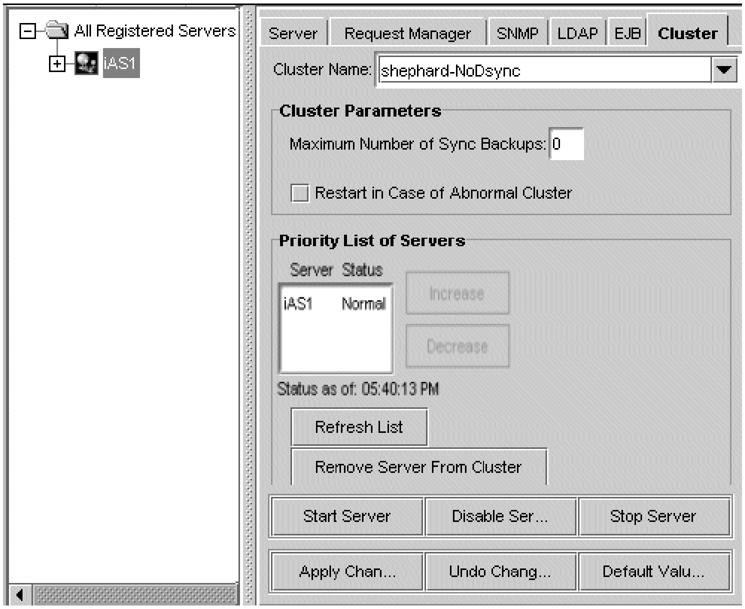
The following sections describe how you can assign or change priority to a sync server:

- To Change Sync Server Priority Using iASAT
- To Change Sync Server Priority Using iPlanet Registry Editor

To Change Sync Server Priority Using iASAT

To assign a new Sync Server priority to a server that is in a cluster, perform the following tasks:

1. From the iASAT toolbar, click General to open the General window.
2. In the left pane of the General window, click a server that is a member of the cluster whose Sync Server priority you want to change.
3. In the right pane of the General window, click the Clusters tab. The following window appears:



Cluster Name: shephard-NoDsync

Cluster Parameters

Maximum Number of Sync Backups: 0

□ Restart in Case of Abnormal Cluster

Priority List of Servers

Server	Status
iAS1	Normal

Status as of: 05:40:13 PM

A list of registered servers is displayed in the left pane of the General window. Another list of servers, sorted by priority in a cluster, is displayed in the right pane.

The list also shows the status of each server in the cluster. The status should always be *Normal*. If an abnormal cluster condition exists, the status could be *Dual Primary* or *No Primary*. To ensure that these conditions are corrected, see Setting Cluster Parameters. To check for status, click the Refresh List button. By default, server status is updated every 15 seconds.

4. In the Priority List of Servers box, click the name of the server whose Sync Server priority you want to change.

5. To change the Sync Server priority of the server, click one of the following buttons next to the Priority List of Servers box:

   - Increase to assign a higher priority.
   - Decrease to assign a lower priority.

6. For example, if a server has a Sync Server priority of third in line to take over for the Sync Primary, clicking Increase once changes the priority from third to second. A lower number indicates higher priority. For example, `0` indicates the highest priority and `1` indicates the next level of priority. The lowest priority value is 65,535.

7. Click Apply Changes, when you complete reassigning priorities for the server(s).

8. For changes in Sync Server priority to apply across a cluster, you must restart each server so that all servers are aware of their new priority sequence, relative to one another.

   The changes you made in iASAT will be reflected in iPlanet Registry. You can also change the sync server priority directly in iPlanet Registry. The next section describes how you can do this.

To Change Sync Server Priority Using iPlanet Registry Editor

To change the sync server priority directly in iPlanet Registry using iPlanet Registry Editor, perform the following tasks:

1. Start iPlanet Registry.

   (See About iPlanet Registry Editor)

NOTE You can have multiple instances of iPlanet Application Server on UNIX systems, each with its own iPlanet Registry. Ensure that you edit the iPlanet Registry that belongs to the server instance for you want to map the synchronizer.

2. Open the following key:

   ```
   SOFTWARE\iPlanet\Application
   Server\Clusters\<clustername>\SyncServers
   ```

3. You will see the IP Address and port number entries of the servers that are added to the cluster. Select the sync server whose priority you want to change.

4. Double-click the entry to open it, or, choose `Modify Value` from the Edit menu. The Modify Value dialog box appears.

5. Change the value as required. Note that a lower value indicates higher priority. For example, `0` indicates the highest priority and `1` indicates the next level of priority. The lowest priority value is `65,535`.

6. Restart every server in the cluster, including the one whose priority you just changed. For changes in Sync Server priority to apply across a cluster, you must restart each server so that all servers are aware of their new priority sequence, relative to one another.

Setting Cluster Parameters

For a cluster to function efficiently, you need to set specific parameters that will affect its performance. You can set these parameters using either iASAT or iPlanet Registry Editor. The following sections describe how you can set cluster parameters using either tool:

- To Set Cluster Parameters Using iASAT
- To Set Cluster Parameters Using iPlanet Registry Editor

To Set Cluster Parameters Using iASAT

You can set the following parameters for each cluster, using iASAT:

- Maximum Number of Sync Backups

  You can specify the maximum number of Sync Backups the Sync Primary will use. In clusters that have many servers, specifying the maximum number of Sync Backups allows you to control how many servers are used as backups.

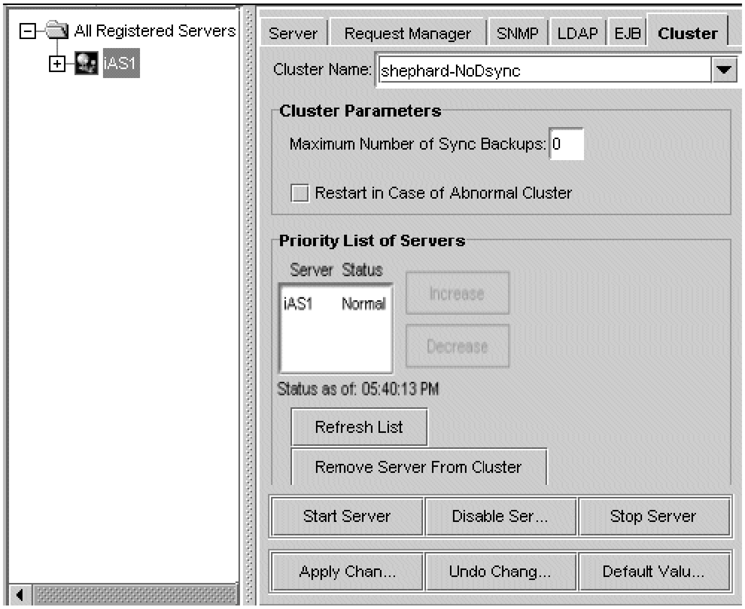
- Restart in case of abnormal cluster

  You can also enable the process to restart if an abnormal cluster condition is detected. An abnormal cluster condition is either a cluster that has more than one iPlanet Application Server instances with the Sync Primary (dual-primary) role or no iPlanet Application Server instances with the Sync Primary role.

To set cluster parameters using iASAT, perform the following steps:

1. From the iASAT toolbar, click General to open the General window.
2. In the left pane of the General window, select the iPlanet Application Server that is a member of the cluster you want to modify.
3. In the right pane of the General window, click the Clusters tab. The following window appears

:



Cluster Name: shephard-NoDsync

Cluster Parameters

Maximum Number of Sync Backups: 0

□ Restart in Case of Abnormal Cluster

Priority List of Servers

Server	Status
iAS1	Normal

Status as of: 05:40:13 PM

4. In the right pane of the General window, enter the maximum number of Sync Backups allowed during a single cluster session in the Maximum Number of Sync Backups text field.
5. Mark the Restart in case of abnormal cluster checkbox to correct any abnormal cluster conditions that are detected.

Restart every server in the cluster. For changes to apply across a cluster, you must restart each instance in the cluster, so that all the instances are aware of the changes.

NOTE You need to first register a server with iASAT, before you add the server to a cluster.

To Set Cluster Parameters Using iPlanet Registry Editor

To set cluster parameters using iPlanet Registry Editor, perform the following steps:

1. Stop the server whose iPlanet Registry you want to edit.

NOTE	Editing the server registry while the server is running can cause serious problems. Also, some changes take effect only after the engine is recycled. It is strongly recommended that you stop all application servers that belong to a cluster, before editing that cluster's settings.

2. Start iPlanet Registry.

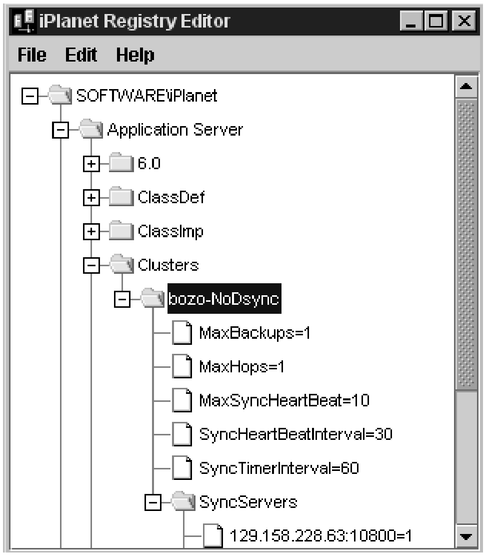
   (See About iPlanet Registry Editor)

NOTE	You can have multiple instances of iPlanet Application Server on UNIX systems, each with its own iPlanet Registry. Ensure that you edit the iPlanet Registry that belongs to the server instance for you want to map the synchronizer.

3. Open the following folder:

   `SOFTWARE\iPlanet\Application Server\Clusters\<clustername>\`

   In the given example, the default cluster is named `bozo-NoDysnc` and contains one Sync Server with a priority of `1`, as shown in the following figure:



```
SOFTWARE\iPlanet
  Application Server
    6.0
    ClassDef
    ClassImp
    Clusters
      bozo-NoDsync
        MaxBackups=1
        MaxHops=1
        MaxSyncHeartBeat=10
        SyncHeartBeatInterval=30
        SyncTimerInterval=60
        SyncServers
          129.158.228.63:10800=1
```

4. Modify the following values under the <`clustername`> key, as required:

TIP To modify a value in iPlanet Registry, perform the following steps:

a. Double-click the key name or select the key and choose Modify Value from the Edit menu, to bring up the Modify Value dialog box.

b. Enter the new value in the dialog box.

c. Click OK to register the change in iPlanet Registry.

- `MaxBackups`

  The maximum number of backup data synchronization servers determines how many Sync Backups are updated with data from the Sync Primary at the same time. For more information about backup data synchronization servers, see What Is a Cluster?.

  As all Sync Backups are updated at the same time, an extra load is created for each additional backup server when you increase the MaxBackups value. Consider the performance impact when you set the number of backup servers, and choose a number that is high enough to provide safety, while not so high as to negatively affect performance. The default value of 1 is usually sufficient.

NOTE The next entry is the `MaxHops` key. This key relates to an unsupported feature. You do not need to modify the value of this key.

- `MaxSyncHeartBeat`

  Check and modify the `MaxSyncHeartBeat` value as required.

  This value specifies the maximum number of heartbeat messages that an engine will send to any other engine. The heart-beat mechanism is used to detect an abnormal cluster condition. The default value is 10.

  Each heartbeat message consists of the:

  - Host ID and port of the engine that sends the messages
  - Role of the sender in the cluster

  Whenever a heartbeat message is received, an iPlanet Application Server engine will send back a response identifying its role in the cluster.

  A heartbeat starts when a Sync Backup server is promoted to a Sync Primary. The new Sync Primary starts to send heart-beat messages to the original Sync Primary engine. In the case of a temporary network failure, the two engines will become Sync Primaries, thus creating a double-primary (split-primary) abnormal condition. This condition can be automatically corrected.

- `SyncHeartBeatInterval`

  Check and modify the `SyncHeartBeatInteveral` value as required.

  This value specifies the number of seconds between two heartbeat messages sent from one server to another. The default value is 30 seconds.

- `SyncTimerInterval`

  Check and modify the `SyncTimerInterval` value as required.

  This key specifies the intervals, in seconds, at which the synchronization service wakes up and checks to see whether any data has expired. This key specifies how often the timer thread goes through the node list and removes all the nodes that have expired.

  If this value is too large, expired data will still be accessible. If this value is too small, the frequent waking up and checking can degrade system performance. The default value of 60 seconds is good for most clusters.

5. Close iPlanet Registry Editor when you finish setting the values you want.
6. Restart all the servers that will be affected by the modifications.

NOTE You can also set cluster parameters using iASAT. See To Set Cluster Parameters Using iASAT for more information.

After correctly completing these steps, you have redefined the default cluster. Now, follow the procedure in “Mapping the Synchronizer to the Cluster” to enable communication between the servers in the cluster

Mapping the Synchronizer to the Cluster

Each iPlanet Application Server has an inbuilt data synchronizer, which helps synchronize the data within the server and also the server’s communication with other servers.

For a cluster to communicate, the synchronizer in each server must know to which cluster the server belongs. If you create clusters using iASAT, the server synchronizer is mapped automatically. If you create clusters manually, using iPlanet Registry Editor, you need to perform the following tasks, to map the synchronizer to its cluster:

1. Stop the server whose iPlanet Registry you want to edit.

NOTE Please be noted that editing the server registry while the server is running can cause serious problems. Also, some changes take effect only after the engine is recycled. It is strongly recommended that you stop all application servers that belong to a cluster, before editing that cluster’s settings.

2. Start iPlanet Registry.

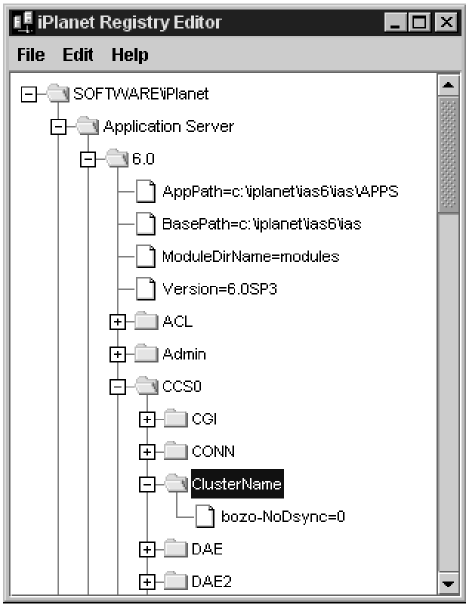
   (See About iPlanet Registry Editor)

NOTE Note that you can have multiple instances of iPlanet Application Server on UNIX systems, each with its own iPlanet Registry. Ensure that you edit the iPlanet Registry that belongs to the server instance for you want to map the synchronizer.

3. Open the following key:

   `SOFTWARE\iPlanet\Application Server\6.0\CCS0\ClusterName\`

   The following window appears:



```
SOFTWARE\iPlanet
  Application Server
    6.0
      AppPath=c:\iplanet\ias6\ias\APPS
      BasePath=c:\iplanet\ias6\ias
      ModuleDirName=modules
      Version=6.0SP3
      ACL
      Admin
      CCS0
        CGI
        CONN
        ClusterName
          bozo-NoDsync=0
        DAE
        DAE2
```

4. The default cluster name will be listed. You can delete the default cluster name and specify a new cluster name for the cluster.

5. Close iPlanet Registry.

6. Start the server whose synchronizer you just mapped.

The synchronizer is now be mapped to the cluster.

How Sync Server Prioritization Improves Coordination

This section discusses general priority issues and gives a comprehensive example of cluster coordination.

Priority values are used only to select between Sync Servers in the same status (either between a group of Sync Backups or between a group of Sync Alternates). Only the order in which instances of iPlanet Application Server are started, not priority, determines which server should be the Sync Primary and which Sync Servers will start out as Sync Backups or Sync Alternates.

A Sync Local is not assigned a priority because it is not eligible to become a Sync Server, so a Sync Local cannot become a Sync Primary, Sync Backup, or Sync Alternate.

Which Sync Server becomes the Sync Primary in a cluster is determined simply by which Sync Server is started first. The next Sync Servers that start, up to the value in `MaxBackups`, become Sync Backups. When the Sync Primary fails, the Sync Backup with the highest priority, which is the lowest integer value, becomes the new Sync Primary.

When a Sync Backup becomes a Sync Primary, the number of Sync Backups falls below the value of `MaxBackups`. To restore the number of Sync Backups, the Sync Alternate with the highest priority becomes a Sync Backup.

Example: Coordination Within a Seven-Server Cluster

The following example illustrates cluster coordination through server roles, and the part that priority plays in determining those roles. As you trace the role changes through the example, keep in mind that server fallibility has been purposely exaggerated to provide many scenarios.

Although not required, you can ease cluster maintenance by assigning the highest priority to the iPlanet Application Server instance that you will start as the Sync Primary, and the next highest priorities (in descending order) to the Sync Backups. Be aware that the cluster in this example does not do this. Also, notice that this cluster does not follow the recommended practice of starting the servers in priority order.

Assume a seven-server cluster with iPlanet Application Server instances that are numbered 0 to 6. Servers 0 through 4 are Sync Servers that are assigned the same priorities as their server numbers (for example, server 0 has a priority of zero). Servers 5 and 6 are Sync Locals. `MaxBackups` for the cluster is set to 2.

- Server 3 is brought up first, so it becomes the Sync Primary.
- Server 4 is started next, and it becomes a Sync Backup.
- Server 6 is started next, and it is a Sync Local.
- Server 1 is started next, and it becomes a Sync Backup.
- Server 2 is started next, and it becomes a Sync Alternate.
- Server 5 is started next, and it is a Sync Local.
- Server 0 is started next, and it becomes a Sync Alternate.

Server 3 fails and goes down. Between the two Sync Backups, server 4 and server 1, server 1 has the higher priority (lower integer value) and it becomes the new Sync Primary. This leaves server 4 as the only Sync Backup.

Because `MaxBackups` is set to 2, one of the Sync Alternates is converted to a Sync Backup. Server 0 becomes the new Sync Backup because it has a higher priority than the other remaining Sync Alternate, server 2. At this point:

- Server 1 is the Sync Primary.
- Servers 0 and 4 are Sync Backups.
- Server 2 is a Sync Alternate.
- Servers 5 and 6 are Sync Locals.
- Server 3 is off-line.

Server 3 comes back online. It becomes a Sync Alternate. Even though it was originally a Sync Primary, the synchronizer now sees it as just another Sync Server, so the server does not resume its Sync Primary role. At this point:

- Server 1 is the Sync Primary.
- Servers 0 and 4 are Sync Backups.
- Servers 2 and 3 are Sync Alternates.
- Servers 5 and 6 are Sync Locals.

Server 0 fails. Server 2 becomes a Sync Backup because it has the higher priority (lower integer value) among the Sync Alternates. At this point:

- Server 1 is the Sync Primary.
- Servers 2 and 4 are Sync Backups.
- Server 3 is a Sync Alternate.
- Servers 5 and 6 are Sync Local servers.
- Server 0 is off-line.

Server 0 comes back online and becomes a Sync Alternate. Server 1, the Sync Primary, fails. Among the Sync Backups, server 2 has a higher priority than server 4, so server 2 becomes the new Sync Primary. Server 0 becomes a Sync Backup. At this point:

- Server 2 is the Sync Primary.
- Servers 0 and 4 are Sync Backups.
- Server 3 is a Sync Alternate.
- Servers 5 and 6 are Sync Locals.
- Server 1 is off-line.

Server 2 fails. Server 0 becomes the Sync Primary and server 3 becomes a Sync Backup. At this point:

- Server 0 is the Sync Primary.
- Servers 3 and 4 are Sync Backups.
- Servers 5 and 6 are Sync Locals.
- Servers 1 and 2 are off-line.

Server 3 fails. Even though only one Sync Backup remains, neither server 5 nor server 6 is considered because neither is a Sync Server. At this point:

- Server 0 is the Sync Primary.
- Server 4 is a Sync Backup.
- Servers 5 and 6 are Sync Locals.
- Servers 1 and 2 and 3 are off-line.

Chapter 14

Setting Up iPlanet Application Server for Development

This chapter describes the tasks you need to perform, to set up iPlanet Application Server for development purposes.

The following subjects are described in this chapter:

- Setting Up Class Paths
- Changing Heap Size For Java Engines
- Changing The Default Application Path
- Enabling Dynamic Servlet Reloading
- Specifying Session And Cache Timeout Values For J2EE Components
- Enabling RMI/IIOP Support
- Setting Environment Variables for Databases

Setting Up Class Paths

You may need to manually set up the iPlanet Application Server class path when you want to do one of the following:

- when you enable RMI/IIOP access to EJBs.
- when you make a set of helper or framework classes available to all applications deployed to a server.

To change the iPlanet Application Server class path, perform the following tasks:

On Solaris

1. Navigate to the `<iasinstall>/ias/env` path and edit the Common Environment script, `iasenv.ksh`. You can use a text editor to edit this file.
2. Search for and locate the `ClassPath` entry. You can either change the default classpath or add a new classpath.
3. Restart the server for the change to take effect.

On Windows

1. Open iPlanet Registry Editor.

   (See About iPlanet Registry Editor)
2. Open the following key:

   `SOFTWARE\iPlanet\Application Server\6.0\Java\`
3. Modify the class path and restart the server for the change to take effect.

Changing Heap Size For Java Engines

Default heap size for KJS is specified in the KJS script. If you want to enable KJS to store more objects, you can change the heap size manually. To do so, perform the following tasks:

On Solaris

1. Go to `<iasinstall>/ias/env` and edit the KJS (Java Engine) startup script, `iasenv.ksh`.

   Locate the `JAVA_ARGS` entry and modify the existing minimum(-*Xms*) and maximum (-*Xmx*) heap size specifications, as required.
2. Close all KJS processes and restart them, for the change in heap size to take effect.

On Windows

1. Start iPlanet Registry Editor.

   (See About iPlanet Registry Editor)

2. Open the following key:

   ```
   SOFTWARE\iPlanet\Application Server\6.0\Java\
   ```

3. Modify the `JavaArgs` value and set the minimum heap size (`-Xms <number>m`) and the maximum heap size (`-Xmx<number>m`) for Java engines.

   Your heap size specification must look like this:

   ```
   JavaArgs=-Xms16m -Xmx32m
   ```

4. Restart all `kjs` processes for the change to take effect.

   Note that on Windows, the heap size for Java engines does not exist by default.

Changing The Default Application Path

Your applications are stored in the default application path, from where they are picked up by DeployTool, to be deployed. You may want to change this path if, for example, you want to store your applications in your server's home directory. You can change the default application through *kregedit*.

Before changing the default application path, you need to:

- create the new path on your machine or network and
- manually copy your application's class files to this location.

After you create the path, you need to change the default class path in Iplanet Registry. To do so, perform the following tasks:

1. Start iPlanet Registry Editor.

   (See About iPlanet Registry Editor)

2. Open the following key:

   ```
   SOFTWARE\iPlanet\Application Server\6.0\
   ```

3. Edit the `AppPath` value and change the default application class path. See Setting Up Class Paths.

4. Restart the server for the change to take effect.

Enabling Dynamic Servlet Reloading

By default, dynamic servlet reloading is disabled in iPlanet Application Server. You can enable it, by performing the following tasks:

1. Start iPlanet Registry Editor.

   (See About iPlanet Registry Editor)

2. Open the following key:

   ```
   SOFTWARE\iPlanet\Application
   Server\6.0\CCSO\SYSTEM_JAVA\Versioning
   ```

3. Set `Disable=0`. The default value is `1` which indicates that dynamic servlet reloading has not been enabled.

4. Restart iPlanet Application Server for the change to take effect.

This change will enable dynamic reloading of servlet classes and registered JSPs (those JSPs that have been assigned GUIDs and are listed as servlets in *web.xm*l files). By default, unregistered JSPs are dynamically reloaded by iPlanet Application Server.

Specifying Session And Cache Timeout Values For J2EE Components

When you deploy an application, session timeout values are set for the applications's J2EE components, such as JSPs, servlets and EJBs. These values apply to all the J2EE components that run on your Web Server and Application Servers.

You can specify session timeout values for individual J2EE components for each application, on a need-basis. The value that you specify for an individual J2EE component will override the default value set during deployment.

Cache timeout values can be specified for servlets and JSPs. These are set to `0` by default when an application is deployed.

You can use DeployTool to set these values too. See the Online Help that has been provided for DeployTool.

This section describes the following topics:

- To Set Session Timeout Value For JSPs And Servlets

- To Set Session Timeout Value For EJBs
- To Set Cache Timeout Settings For JSPs And Servlets

To Set Session Timeout Value For JSPs And Servlets

For JSPs and Servlets, session timeout value can be set for each application. These values are set by default during deployment. You can modify the session timeout value, by performing the following tasks:

1. Start iPlanet Registry Editor.

   (See About iPlanet Registry Editor)

2. Open the following key:

   ```
   SOFTWARE\iPlanet\Application Server\6.0\J2EE-Module\<module
   name>\
   ```

   If session timeout value has been specified already, you will see a key that looks like this: `session-timeout=()`. If session timeout value has not been specified, you can add it to the module.

3. Select the module for which you want to specify session timeout value.
4. Click `Edit>Add Value`.
5. In the Name field, specify a name, for example, `Session-timeout`.
6. In the Value field, specify a value, for example, *10*. Note that the timeout value is specified in seconds.
7. Set the type of the value to `Integer`.
8. Click OK to save your changes.

The session timeout value is set for the Servlets and JSPs that belong to your application.

To Set Session Timeout Value For EJBs

For EJBs, session timeout value is globally set using iASAT. See To Specify EJB Container Parameters for Run Time for more information. The value that is set through iASAT is applicable to all the EJBs that are deployed.

However, you can customize the session timeout value for each individual EJB, in iPlanet Registry. To specify the session timeout value for an EJB, perform the following tasks:

1. Start iPlanet Registry Editor.

   (See About iPlanet Registry Editor)

2. Open the following key:

   ```
   SOFTWARE\iPlanet\Application Server\6.0\J2EE-Module\<module
   name>\ejbs
   ```

   You will see a string that looks like this:

   ```
   <Beanhomename>=(18c06cf0-21dd-11d2094b6-0060083a5082)
   ```

   The string within brackets is the GUID (Global Unique Identifier) that identifies the EJB used in the selected application module. Make a note of this GUID.

3. Navigate to `SOFTWARE\iPlanet\Application Server\ClassDef\`

   This folder lists all the GUIDS that are attached to deployed applications. Locate the GUID you want.

4. Expand the GUID key into its hierarchal tree.

5. Open the `SessionDescriptor` key. This key contains session timeout, passivation timeout and other values that have been set for the EJB.

6. Modify the session timeout value and restart the server for the change to take effect.

To Set Cache Timeout Settings For JSPs And Servlets

The default cache timeout is the period for which servlets and JSPs are kept in the cache before they are purged by the Web Server. The default cache time is set at the time of deployment.

You can change the default cache timeout settings for JSPs and Servlets, per GUID. To do so, perform the following tasks:

1. Start iPlanet Registry Editor.

   (See About iPlanet Registry Editor)

2. Open the following key:

   `SOFTWARE\iPlanet\Application Server\6.0\J2EE-Module\<module name>\servlets\`

   You will see a GUID, which looks like this:

   `<module name>servlet=(1a488137-7510-1941-bae5-080020b90148)`

   Make a note of this GUID.

3. Navigate to `SOFTWARE\iPlanet\Application Server\ClassDef\`

   This folder lists all the GUIDS that are attached to deployed applications. Locate the GUID you want.

4. Select the GUID and click `Edit>Add Key`.

5. In the Name field, type `Caching`.

6. Create the following values under this key:

   - Cache-Size (value in KB, type `Integer`)
   - Cache-Timeout (value in seconds, type `Integer`)
   - Cache-Option (value can be set as either `Timeout-Create` or `Timeout-Lastaccess`, type `String`).
   - Cache-Criteria (value that specifies appropriate cache criteria, type `String`). See the Java Developer's Guide for more information on cache criteria.

7. Restart the server for the changes to take effect.

If your application uses JSPs, you will find the relevant GUID in the path described in Step 2.

Enabling RMI/IIOP Support

iPlanet Application Server does not enable support for RMI/IIOP access to EJBs, during installation. However, you can add an RMI/IIOP bridge process to the iPlanet Application Server environment through iASAT.

To add an RMI/IIOP bridge process to the iPlanet Application Server environment through iASAT, perform the following tasks:

1. Start iASAT.
   - On Solaris, go to `<iASinstall>/ias/bin/` and type `ksvradmin`.
   - On Windows, from the Start menu, choose `Programs>iPlanet Application Server` and choose iAS Administration Tool.
2. Select the server for which you want to add a new RMI/IIOP process. Expand the server to view the processes under it.

   You should see at least one kjs and one kxs process. You need to now add a cxs process, to enable RMI/IIOP process.
3. Click `File>New`. Choose Process.
4. Select `cxs` from the Process drop down list.
5. In the Port text field, enter a port number that is different from the port numbers being used by the other processes.
6. In the IIOP Port text field, specify a port number for the IIOP bridge process. The default IIOP port number is 9010. You can retain this port number if it does not conflict with the other port numbers already in your system environment.
7. Click OK to register the cxs process. The cxs process is added to the iPlanet Application Server machine.
8. Restart the server to activate the cxs process.

On Solaris, you can also check for the existence of the IIOP bridge process from the command line. For example:

```
ps - ef |grep iiop
```

```
root 1153 1 0 17:00:15 ? 0:00 /bin/sh/usr/iPlanet/iAS6/bin/kjs -cset
CCSO - eng 3 - iiop - DORBinsPOrt=9010
```

Setting Environment Variables for Databases

Although system environment variables for databases are set during installation, verify that they have not changed.

On Solaris, you can type env at the prompt, to verify the environment variables. Review the list of environment variables. If a any variable in the system environment is not set according to the guidelines provided here, change it to the proper setting.

Sybase

- For Bourne Shell:

  ```
  DSQUERY=<sybase servername>; export DSQUERY
  ```

- For C Shell:

  ```
  setenv DSQUERY <sybase servername>
  ```

  Replace `sybase servername` with the name of the user's Sybase server.

Oracle

- For Bourne Shell:

  ```
  ORACLE_HOME=<oracle install directory>; export ORACLE_HOME
  ```

- For C Shell

  ```
  setenv ORACLE_SID <oracle SID>
  ```

  ```
  setenv ORACLE_HOME <oracle install directory>
  ```

DB2

- For Bourne Shell:

  ```
  DB2INSTANCE=<db2instance>; export DB2INSTANCE
  ```

- For C Shell

  ```
  setenv DB2INSTANCE <db2instance>
  ```

INFORMIX

- For Bourne Shell:

  ```
  INFORMIXSERVER=<informixserver>; export INFORMIXSERVER
  ```

- For C Shell

  ```
  setenv INFORMIXSERVER <informixserver>
  ```

Adjusting Environment Size

After installation, if you notice any Administrative Server processes consuming 100% of your system resources, increase the size of the environment space on your NT machine. Edit the file `Config.nt`, found in the System32 directory of your system root (normally `C:\Winnt` or `C:\Winnt4`) by adding the following line, and then restart your machine:

```
SHELL=%systemroot%\system32\command.com /e:2048
```

If you still experience the problem after restarting your machine, try increasing the environment size further by specifying `/e:4096`, instead of `/e:2048`, in the line above.

Appendix A

Troubleshooting

This appendix contains the following information about troubleshooting iPlanet Application Server:

- Configuring the Class Path
- Setting up Transactions
- What Is a Lock Held by In-Doubt Error?

Configuring the Class Path

When running applications, if the iPlanet Application Server Class Loader is unable to find the AppLogic class file through the `SYSTEM_JAVA` parameter (the registry parameter that contains both the CLASSPATH and GX_CLASSPATH settings) in the registry, iPlanet Application Server hands the request over to the Java Class Loader, which in turn reads the `CLASSPATH` environment variable to find the class file. This allows AppLogics and servlets to execute even if the user class path is not specified.

Setting up Transactions

When configuring your resource manager for use in global transactions, you might encounter one or more of the following problems:

- What if xa_open Fails?
- What if xa_recover Fails?
- What Is a Lock Held by In-Doubt Error?

- How Do I Configure the Number of Server-Side Connections?

What if xa_open Fails?

If an xa_open failure message appears in your log file, you may have a problem with the open string. Global connections rely on open strings, which provide information for global transaction initialization. When installing iPlanet Application Server, the installation program puts default values in this open string. Check to be sure that the server name, user name and password are set correctly. See Setting Up Resource Managers for Distributed Transactions for the appropriate open string format for your database.

If you find an XAER_RMERR error, you have set the server instance incorrectly in the open string or the server is down.

If you find an XAER_INVAL error, there is a syntax error in your open string.

What if xa_recover Fails?

The following is an example of an xa_recover failure:

```
1 00271 99/04/30-10:00:28.124250 5c2c0837 W  xa_recover to RM 0
returned x tCode -- 0xfffffffd (XAER_RMERR)
```

```
1 00271 99/04/30-10:00:28.124250 5c3c1017 W  Terminating recovery
scan for  0.
```

An xa_recover failure indicates that the database server is not set up for recovery. You must run the appropriate database setup script to create recovery tables and procedures.

For example, for Oracle databases, run the following scripts with sys permissions from the sqlplus prompt:

```
ftp://ftp1.iplanet.com/private/ias/60beta2/extra/xa_sql/xaviews.sql
```

```
ftp://ftp1.iplanet.com/private/ias/60beta2/extra/xa_sql/xaviews_add
.sql
```

What Is a Lock Held by In-Doubt Error?

Global transactions are left “hanging” or in-doubt when a Java Server (KJS) process is abruptly killed or crashes. When the KJS process restarts, these transactions are rolled back, but if you want to manually delete them, refer to Resolving In-Doubt Transactions.

How Do I Configure the Number of Server-Side Connections?

Once a global transaction is started on a thread, a connection is tied to that thread. Therefore, when configuring the number of server-side connections, use the total number of Java Server (KJS) threads in your enterprise.

For example, for Oracle databases, change the value of `max_number_processes` in the `initinstancename.ora` file in the `pfile` directory of the Oracle server installation.

Glossary

Access Control Entries (ACEs) A hierarchy of rules which the web server uses to evaluate incoming access requests.

Access Control List (ACL) A collection of ACEs. An ACL is a mechanism for defining which users have access to your server. You can define ACL rules that are specific to a particular file or directory, granting or denying access to one or more users and groups.

admpw The username and password file for the Enterprise Administrator Server superuser.

agent Software that runs the network-management software in a network device, such as a router, host, or X terminal. See also intelligent agents.

authentication Allows client to verify that they are connected to an SSL-enabled server, preventing another computer from impersonating the server or attempting to appear SSL-enabled when it isn't.

authorization The granting of access to an entire server or particular files and directories on it. Authorization can be restricted by criteria including hostnames and IP addresses.

browser See client.

cache A copy of original data that is stored locally. Cached data doesn't have to be retrieved from a remote server again when requested.

certification authority (CA) A third-party organization that issues digital files used for encrypted transactions.

certificate A nontransferable, nonforgeable, digital file issued from a third party that both communicating parties already trust.

Certificate revocation list (CRL) CA list, provided by the CA, of all revoked certificates.

Compromised key list (CKL) A list of key information about users who have compromised keys. The CA also provides this list.

CGI Common Gateway Interface. An interface by which external programs communicate with the HTTP server. Programs that are written to use CGI are called CGI programs or CGI scripts. CGI programs handle forms or parse output the server does not normally handle or parse.

chroot `An additional` root directory ypu can create to limit the server to specific directories. You'd use this feature to safeguard an unprotected server.

cipher A cipher is a cryptographic algorithm (a mathematical function), used for encryption or decryption.

ciphertext Information disguised by encryption, which only the intended recipient can decrypt.

client Software, such as Netscape Navigator, used to request and view World Wide Web material. Also known as a browser program.

client auth Client authentication.

collection A database that contains information about documents, such as word list and file properties. Collections are used by the search function to retrieve documents matching specified search criteria.

Common LogFile Format The format used by the server for entering information into the access logs. The format is the same among all major servers, including the iPlanet FastTrack and Enterprise servers.

connection group.

DHCP Dynamic Host Configuration Protocol. An Internet Proposed Standard Protocol that allows a system to dynamically assign an IP address to individual computers on a network.

daemon (Unix) A background process responsible for a particular system task.

DNS Domain Name System. The system that machines on a network use to associate standard IP addresses (such as 198.93.93.10) with hostnames (such as `www.netscape.com`). Machines normally get this translated information from a DNS server, or they look it up in tables maintained on their systems.

DNS alias A hostname that the DNS server knows points to a different host—specifically a DNS CNAME record. Machines always have one real name, but they can have one or more aliases. For example, an alias such as `www.`*yourdomain.domain* might point to a real machine called `realthing.`*yourdomain.domain* where the server currently exists.

document root A directory on the server machine that contains the files, images, and data you want to present to users accessing the server.

drop word See stop word.

encryption The process of transforming information so it can't be decrypted or read by anyone but the intended recipient.

Administration Server A web-based server that contains the forms you use to configure all of your iPlanet Web Servers.

expires header The expiration time of the returned document, specified by the remote server.

extranet An extension of a company's intranet onto the Internet, to allow customers, suppliers, and remote workers access to the data.

fancy indexing A method of indexing that provides more information than simple indexing. Fancy indexing displays a list of contents by name with file size, last modification date, and an icon reflecting file type. Because of this, fancy indexes might take longer than simple indexes for the client to load.

file extension The last part of a filename that typically defines the type of file. For example, in the filename `index.html` the file extension is `html`.

file type The format of a given file. For example, a graphics file doesn't have the same file type as a text file. File types are usually identified by the file extension (`.gif` or `.html`).

firewall A network configuration, usually both hardware and software, that protects networked computers within an organization from outside access. Firewalls are commonly used to protect information such as a network's email and data files within a physical building or organization site.

flexible log format A format used by the server for entering information into the access logs.

FORTEZZA An encryption system used by U.S. government agencies to manage sensitive but unclassified information.

FTP File Transfer Protocol. An Internet protocol that allows files to be transferred from one computer to another over a network.

GIF Graphics Interchange Format. A cross-platform image format originally created by CompuServe. GIF files are usually much smaller in size than other graphic file types (BMP, TIFF). GIF is one of the most common interchange formats. GIF images are readily viewable on Unix, Microsoft Windows, and Apple Macintosh systems.

hard restart The termination of a process or service and its subsequent restart. See also soft restart.

home page A document that exists on the server and acts as a catalog or entry point for the server's contents. The location of this document is defined within the server's configuration files.

hostname A name for a machine in the form *machine.domain.dom*, which is translated into an IP address. For example, `www.iplanet.com` is the machine `www` in the subdomain `iplanet` and `com` domain.

HTML Hypertext Markup Language. A formatting language used for documents on the World Wide Web. HTML files are plain text files with formatting codes that tell browsers such as the Netscape Navigator how to display text, position graphics and form items, and display links to other pages.

HTTP HyperText Transfer Protocol. The method for exchanging information between HTTP servers and clients.

HTTP-NG The next generation of HyperText Transfer Protocol.

HTTPD An abbreviation for the HTTP daemon or service, a program that serves information using the HTTP protocol. The Netscape Enterprise Server is often called an HTTPD.

HTTPS A secure version of HTTP, implemented using the Secure Sockets Layer, SSL.

imagemap A process that makes areas of an image active, letting users navigate and obtain information by clicking the different regions of the image with a mouse. Imagemap can also refer to a CGI program called “imagemap,” which is used to handle imagemap functionality in other HTTPD implementations.

inittab (Unix) A Unix file listing programs that need to be restarted if they stop for any reason It ensures that a program runs continuously. Because of its location, it is also called `/etc/inittab`. This file isn’t available on all Unix systems.

intelligent agent An object within a server that performs various requests (such as HTTP, NNTP, SMTP, and FTP requests) on behalf of the user. In a sense, the intelligent agent acts as a client to the server, making requests that the server fulfills.

IP address Internet Protocol address. A set of numbers, separated by dots, that specifies the actual location of a machine on the Internet (for example, 198.93.93.10).

ISDN Integrated Services Digital Network.

ISINDEX An HTML tag that turns on searching in the client. Documents can use a network navigator’s capabilities to accept a search string and send it to the server to access a searchable index without using forms. In order to use <ISINDEX>, you must create a query handler.

ISMAP ISMAP is an extension to the IMG SRC tag used in an HTML document to tell the server that the named image is an imagemap.

ISP Internet Service Provider. An organization that provides Internet connectivity.

Java An object-oriented programming language created by Sun Microsystems used to create real-time, interactive programs called applets.

JavaScript A compact, object-based scripting language for developing client and server Internet applications.

JavaServer Pages Extensions that enable all JavaServer page metafunctions, including instantiation, initialization, destruction, access from other components, and configuration management. JavaServer pages, are reusable Java applications that run on a web server rather than in a web browser.

Java Servlets Extensions that enable all Java servlet metafunctions, including instantiation, initialization, destruction, access from other components, and configuration management. Java servlets are reusable Java applications that run on a web server rather than in a web browser.

last-modified header The last modification time of the document file, returned in the HTTP response from the server.

magnus.conf The main Enterprise Server configuration file. This file contains global server configuration information (such as, port, security, and so on). This file sets the values for variables that configure the server during initialization. Enterprise Sever reads this file and executes the variable settings on startup. The server does not read this file again until it is restarted, so you must restart the server every time you make changes to this file.

MD5 A message digest algorithm by RSA Data Security. MD5 can be used to produce a short digest of data that is unique with high probability. It is mathematically extremely hard to produce a piece of data that produces the same message digest email.

MD5 signature A message digest produced by the MD5 algorithm.

MIB Management Information Base.

MIME Multi-Purpose Internet Mail Extensions. An emerging standard for multimedia email and messaging.

mime.types The MIME (Multi-purpose Internet Mail Extension) type configuration file. This file maps file extensions to MIME types, to enable the server to determine the type of content being requested. For example, requests for resources with `.html` extensions indicate that the client is requesting an HTML file, while requests for resources with `.gif` extensions indicate that the client is requesting an image file in GIF format.

MTA Message Transfer Agent. You must define your server’s MTA Host to use agent services on your server.

Netscape Console A Java application that provides server administrators with a graphical interface for managing all Netscape servers from one central location anywhere within your enterprise network. From any installed instance of Netscape Console, you can see and access all the Netscape servers on your enterprise's network to which you have been granted access rights.

NIS (Unix) Network Information Service. A system of programs and data files that Unix machines use to collect, collate, and share specific information about machines, users, file systems, and network parameters throughout a network of computers.

network management station (NMS) A machine users can use to remotely manage a network. A managed device is anything that runs SNMP such as hosts, routers, and Netscape/iPlanet servers. An NMS is usually a powerful workstation with one or more network management applications installed.

NNTP Network News Transfer Protocol for newsgroups. You must define your news server host to use agent services on your server.

NSAPI See Server Plug-in API.

obj.conf The server's object configuration file. This file contains additional initialization information, settings for server customization, and instructions that the server uses to process requests from clients (such as browsers). Enterprise Server reads this file every time it processes a client request.

password file (Unix) A file on Unix machines that stores Unix user login names, passwords, and user ID numbers. It is also known as `/etc/passwd`, because of where it is kept.

primary document directory See document root.

protocol A set of rules that describes how devices on a network exchange information.

private key The decryption key used in public-key encryption.

public key The encryption key used in public-key encryption.

public information directories (Unix) Directories not inside the document root that are in a Unix user's home directory, or directories that are under the user's control.

Quality Feedback Agent An error-handling mechanism that enables you to automatically send error information (stack and register dump) to Netscape.

RAM Random access memory. The physical semiconductor-based memory in a computer.

rc.2.d (Unix) A file on Unix machines that describes programs that are run when the machine starts. This file is also called `/etc/rc.2.d` because of its location.

redirection A system by which clients accessing a particular URL are sent to a different location, either on the same server or on a different server. This system is useful if a resource has moved and you want the clients to use the new location transparently. It's also used to maintain the integrity of relative links when directories are accessed without a trailing slash.

resource Any document (URL), directory, or program that the server can access and send to a client that requests it.

RFC Request For Comments. Usually, procedures or standards documents submitted to the Internet community. People can send comments on the technologies before they become accepted standards.

root (Unix) The most privileged user on Unix machines. The root user has complete access privileges to all files on the machine.

server daemon A process that, once running, listens for and accepts requests from clients.

Server Plug-in API An extension that allows you to extend and / or customize the core functionality of Netscape servers and provide a scalable, efficient mechanism for building interfaces between the HTTP server and back-end applications. Also known as NSAPI.

server root A directory on the server machine dedicated to holding the server program, configuration, maintenance, and information files.

simple index The opposite of fancy indexing—this type of directory listing displays only the names of the files without any graphical elements.

SNMP Simple Network Management Protocol.

SOCKS Firewall software that establishes a connection from inside a firewall to the outside when direct connection would otherwise be prevented by the firewall software or hardware (for example, the router configuration).

soft restart A way to restart the server that causes the server to internally restart, that is, reread its configuration files. A soft restart sends the process the `HUP` signal (signal number one). The process itself does not die, as it does in a hard restart.

SSL Secure Sockets Layer. A software library establishing a secure connection between two parties (client and server) used to implement HTTPS, the secure version of HTTP.

stop word A word identified to the search function as a word not to search on. This typically includes such words as *the, a, an, and.* Also referred to as *drop words.*

strftime A function that converts a date and a time to a string. It's used by the server when appending trailers. `strftime` has a special format language for the date and time that the server can use in a trailer to illustrate a file's last-modified date.

superuser (Unix) The most privileged user available on Unix machines (also called root). The superuser has complete access privileges to all files on the machine.

Sym-links (Unix) Abbreviation for symbolic links, which is a type of redirection used by the Unix operating system. Sym-links let you create a pointer from one part of your file system to an existing file or directory on another part of the file system.

TCP/IP Transmission Control Protocol/Internet Protocol. The main network protocol for the Internet and for enterprise (company) networks.

telnet A protocol where two machines on the network are connected to each other and support terminal emulation for remote login.

timeout A specified time after which the server should give up trying to finish a service routine that appears hung.

TLS Secure Sockets Layer. A software library establishing a secure connection between two parties (client and server) used to implement HTTPS, the secure version of HTTP.

top (Unix) A program on some Unix systems that shows the current state of system resource usage.

top-level domain authority The highest category of hostname classification, usually signifying either the type of organization the domain is (for example, `.com` is a company, `.edu` is an educational institution) or the country of its origin (for example, `.us` is the United States, `.jp` is Japan, `.au` is Australia, `.fi` is Finland).

uid (Unix) A unique number associated with each user on a Unix system.

URI Uniform Resource Identifier. A file identifier that provides an additional layer of security by using an abbreviated URL. The first part of the URL is substituted with a URL mapping that hides the file's full physical pathname from the user. See also URL mapping.

URL Uniform Resource Locator. The addressing system used by the server and the client to request documents. A URL is often called a location. The format of a URL is *protocol://machine:port/document*.

A sample URL is `http://www.netscape.com/index.html`.

URL database repair A process that repairs and updates a URL database that has been damaged by a software failure, a system crash, a disk breakdown, or a full file system.

URL mapping The process of mapping a document directory's physical pathname to a user-defined alias so that files within the directory need only refer to the directory's alias instead of the file's full physical pathname. Thus, instead of identifying a file as `usr/Netscape/SuiteSpot/docs/index.html`, you could identify the file as `/myDocs/index.html`. This provides additional security for a server by eliminating the need for users to know the physical location of server files.

web publishing The capability of server clients to access and manipulate server files, editing and publishing documents remotely. Web publishing provides document version control, link management, search, access control, and agent services to server users.

web application A collection of servlets, JavaServer Pages, HTML documents, and other web resources which might include image files, compressed archives, and other data. A web application may be packaged into an archive (a WAR file) or exist in an open directory structure.

Web Application Archive (WAR) An archive file that contains a complete web application in compressed form. iPlanet Web Server cannot access an application in a WAR file. You must uncompress a web application (deploy it using the `wdeploy` utility) before iPlanet Web Server can serve it.

Windows CGI (Windows NT) CGI programs written in a Windows-based programming language such as Visual Basic.

Index