Administration Guide

iPlanet Application Server

Version 6.0, SP2

806-3493-01
January 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

Preface

This preface contains the following topics:

- Using the Documentation
- 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 and iPlanet Application Builder (iAB) printed manuals and online read-me file. 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 as online files in PDF and HTML format.

Table 1 Where to Find Information

For information about	See the following	Shipped with
Late-breaking information about the software and the documentation	`readme.htm`	iPlanet Application Server 6.0 on Solaris / NT, iAB 6.0 on Solaris / NT
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 on Solaris / NT
Installing iPlanet Application Builder.	`install.htm`	iAB 6.0 on Solaris / NT

Table 1 Where to Find Information *(Continued)*

For information about	See the following	Shipped with
Basic features of iPlanet Application Server, such as its software components, general capabilities, and system architecture.	*Overview*	iPlanet Application Server 6.0 on Solaris / NT, iAB 6.0 on Solaris / NT
Administering one or more application servers using the iPlanet Application Server Administration Tool to perform the following tasks: • Monitoring and logging server activity • Setting up users and groups • Administering database connectivity • Administering transactions • Load balancing servers • Managing distributed data synchronization	*Administration Guide*	iPlanet Application Server 6.0 on Solaris / NT
Deploying J2EE applications to iPlanet Application Server using the Deployment Tool.	*JavaHelp* (integrated into the iPlanet Application Server Deployment Tool)	iPlanet Application Server 6.0 on Solaris / NT
Migrating your applications to the new iPlanet Application Server 6.0 programming model from version 4.0, including a sample migration of an Online Bank application provided with iPlanet Application Server	*Migration Guide*	iPlanet Application Server 6.0 on Solaris / NT, iAB 6.0 on Solaris / NT

Table 1 Where to Find Information *(Continued)*

For information about	See the following	Shipped with
Creating iPlanet Application Server 6.0 applications within an integrated development environment by performing the following tasks: • Creating and managing projects • Using wizards • Creating data-access logic • Creating presentation logic and layout • Creating business logic • Compiling, testing, and debugging applications • Deploying and downloading applications • Working with source control • Using third-party tools	*User's Guide*	iAB 6.0 on Solaris / NT

Table 1 Where to Find Information *(Continued)*

For information about	See the following	Shipped with
Creating iPlanet Application Server 6.0 applications that follow the new 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	*Programmer's Guide (Java)*	iPlanet Application Server 6.0 Solaris / NT, iAB 6.0 Solaris / NT
Using the public classes and interfaces, and their methods in the iPlanet Application Server class library to write Java applications	*Server Foundation Class Reference (Java)*	iPlanet Application Server 6.0 on Solaris / NT, iAB 6.0 on Solaris / NT
Creating iPlanet Application Server C++ applications using the iPlanet Application Server class library by performing the following tasks: • Designing applications • Writing AppLogics • Creating HTML templates • Creating queries • Running and debugging applications	*Programmer's Guide (C++)*	Order separately

Table 1 Where to Find Information *(Continued)*

For information about	See the following	Shipped with
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 *Administration Guide* guide leads you through the tasks that you perform as the administrator of one or more iPlanet Application Server machines. This guide assumes 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" on page 20.

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 the material in Part I, "Getting Started" 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 information specific to a multiple-server enterprise.

Part I: Getting Started

The first part of the *Administration Guide* describes the environment of iPlanet Application Server.

The following chapter is included in this part:

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

Part II: Administering a Single iPlanet Application Server

The second part of the *Administration Guide* describes server and application administration procedures for a single iPlanet Application Server machine. The procedures included in this part are those that you are most likely to do right away.

The following chapters are included in this part:

- Chapter 2, “Monitoring Server Activity,” describes the monitoring service provided by iPlanet Application Server Administration Tool 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 Applications,” describes how to set up users and groups to provide security for your applications.
- Chapter 6, “Increasing Fault Tolerance and 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 *Administration Guide* describes how to administer multiple iPlanet Application Server machines. Included are more in-depth administration procedures and concepts that apply to a multiple-server enterprise. 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.
- Appendix A, "Troubleshooting," contains troubleshooting information about your iPlanet Application Server machine.

Documentation Conventions

File and directory paths are given in Windows format (with backslashes separating directory names). For Unix versions, the directory paths are the same, except slashes are used instead of backslashes to separate directories.

This guide uses URLs of the form:

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

In these URLs, *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; and *file* is an individual filename. Italic items in URLs are placeholders.

This guide uses the following font conventions:

- 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, emphasis, variables and placeholders, and words used in the literal sense.

Part 1

Getting Started

Chapter 1

Performing Basic Tasks with the Administration Tool

This chapter describes how to get started with iPlanet Application Server Administration Tool, as well as the basic iPlanet Application Server configuration tasks you can perform using either iPlanet Application Server Administration Tool or at the command line.

The following topics are included in this chapter:

- About iPlanet Application Server Administration Tool
- Starting the Administration Tool
- Registering an iPlanet Application Server
- Unregistering a Server
- Starting iPlanet Application Server From the Administration Tool
- Setting EJB Container Parameters for Run Time
- Using the iPlanet Registry Editor
- Updating the Installation Key
- Changing the IP Address

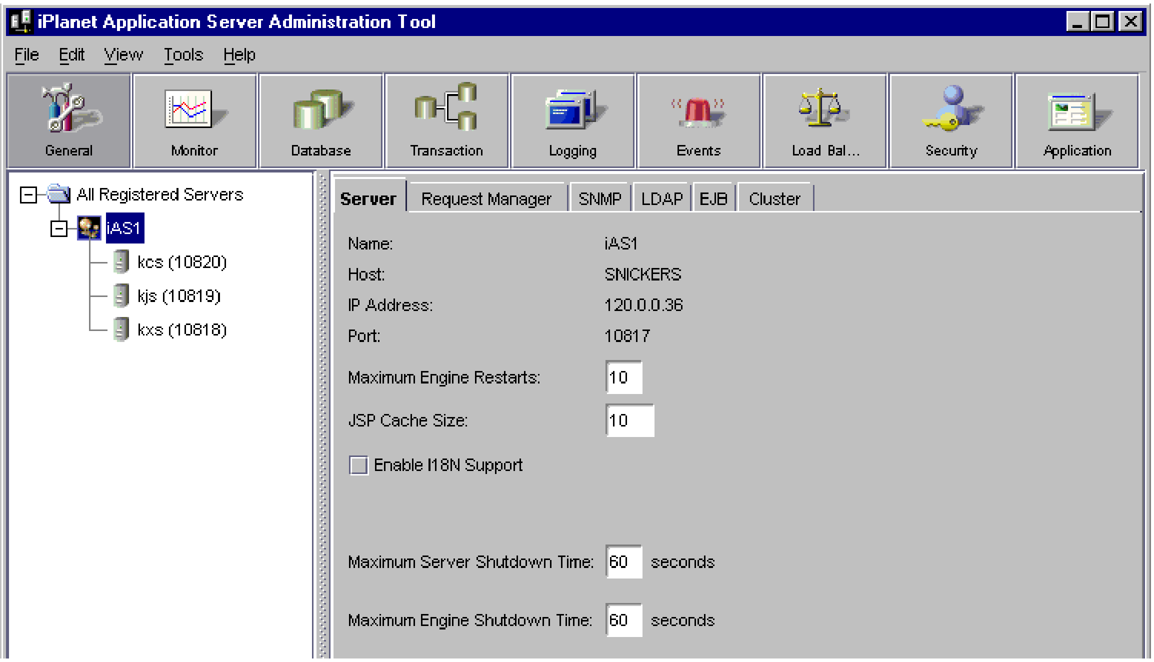
About iPlanet Application Server Administration Tool

iPlanet Application Server Administration Tool 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 such performance-related tasks as adjusting database connection threads and load-balancing parameters. Server administrators must also separately configure components the application server uses, including the web server.

You also use the iPlanet Application Server Administration Tool to administer application components. Application administration involves managing application components by grouping, enabling, and partitioning them to achieve better application performance. Application components, the core of an iPlanet Application Server application, are stored on the application server and contained in code written by the application developer. 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 *Programmer's Guide*.

Administrative tasks are all performed using iPlanet Application Server Administration Tool. The left panel of the iPlanet Application Server Administration Tool's main window displays all iPlanet Application Server machines registered with the Administration Tool. The right panel of that window displays individual features.

When iPlanet Application Server Administration Tool 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: SNICKERS

IP Address: 120.0.0.36

Port: 10817

Maximum Engine Restarts: 10

JSP Cache Size: 10

□ Enable I18N Support

Maximum Server Shutdown Time: 60 seconds

Maximum Engine Shutdown Time: 60 seconds

Starting the Administration Tool

To administer one or more iPlanet Application Server machines, start the iPlanet Application Server Administration Tool by performing one of the following tasks:

- On Windows NT system: from the Start menu, choose Programs, then choose iPlanet Application Server 6.0. Finally, choose iPlanet Application Server Administration Tool.
- On a UNIX system: First make sure the PATH variable contains the absolute path to the iPlanet Application Server bin directory and then type the following at the command prompt:

  ```
  ksvradmin &
  ```

Registering 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.

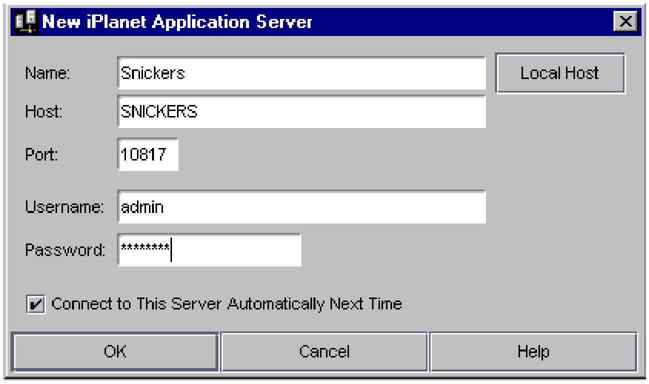
iPlanet Application Server must be registered before you can manage it with the Administration Tool.

To register iPlanet Application Server, perform the following steps:

1. On the iPlanet Application Server Administration Tool toolbar, click the General button to open the General window.

2. From the File menu, choose New, then 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.

   - 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 by default to port 10817.

6. In the User Name and Password text box, specify the user name and password you entered during installation of the server or when modifying the Users and Groups.

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

8. Click OK to register the server.

Unregistering a Server

You can remove a server from the scope of the enterprise when that server is no longer available.

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

1. On the iPlanet Application Server Administration Tool toolbar, click the General button 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 or servers you want to delete.

4. From the Edit menu, choose Delete.

   The selected server is removed from the scope of the iPlanet Application Server Administration Tool.

Starting iPlanet Application Server From the Administration Tool

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 manually start the server by performing the following steps:

1. Click the General button on the iPlanet Application Server Administration Tool 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. Note that you can not expand the servers in the hierarchical tree when they are not running.

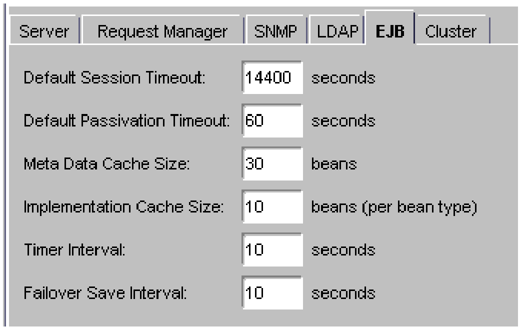
Setting EJB Container Parameters for Run Time

iPlanet Application Server provides an Enterprise Java Bean (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 iPlanet Application Server Administration Tool.

To access the editor, perform the following steps:

1. On the iPlanet Application Server Administration Tool toolbar, click the General button 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. Applies to stateful session EJBs.
   - Default Passivation Timeout: time in seconds that elapses before the state of the EJB, which is currently in memory, is written to disk. This value must be less than session timeout.
   - Metadata Cache Size: refers to the metadata cache for EJBs. Value is in number of EJBs.

- Implementation Cache Size: maximum cache size 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. When you are finished setting the EJB container parameters, click Apply Changes.

   You must restart the server before changes take effect.

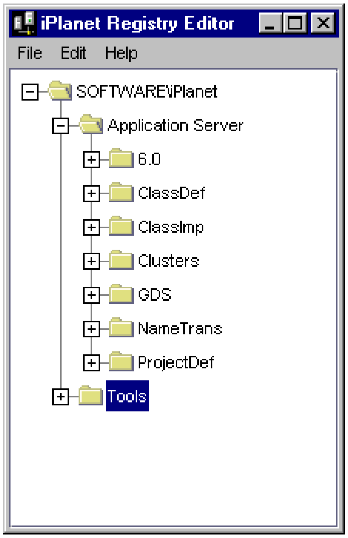
Using the iPlanet Registry Editor

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 machines. You should always use the iPlanet Registry Editor, instead of `regedit`, to manage registry entries for iPlanet Application Server as it displays values stored not only in your local machine's registry, but in your Directory Server as well.

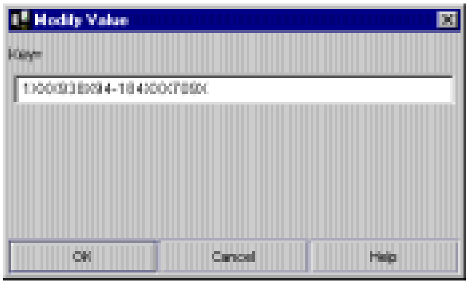
You can launch the iPlanet Registry Editor by typing `kregedit` at the command line on Solaris machines.

For Windows NT machines, click the Windows NT Start button and choose Run. Type `kregedit` and click OK.

The following window appears:



To modify a value in the registry, double-click the entry. A dialog box similar to the following appears:



Key=

1XXX338X94-184XXX705XX

Updating the Installation Key

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

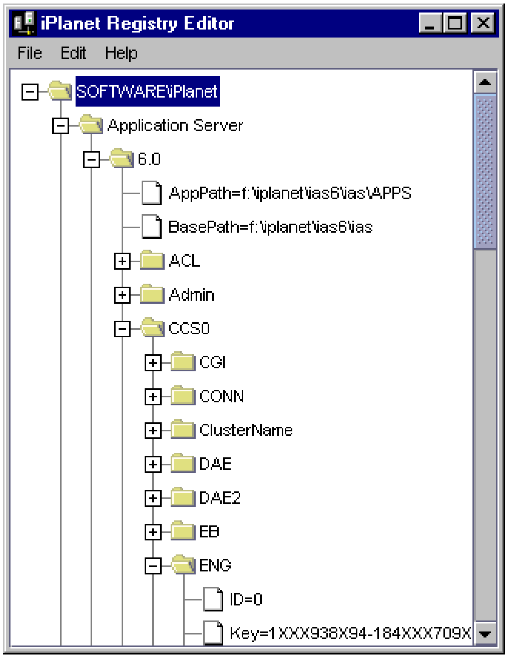
To reset the installation key, perform the following steps:

1. Shutdown iPlanet Application Server.

2. Open the iPlanet Registry Editor by typing `kregedit` at the command line.

   (See “Using the iPlanet Registry Editor” on page 25.)

   The following window appears:

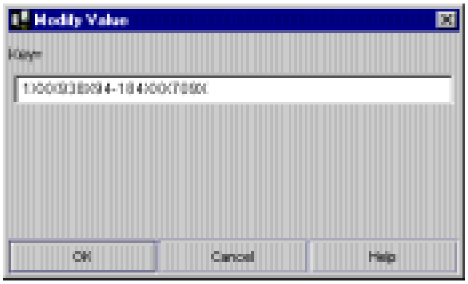


```
SOFTWARE\iPlanet
    Application Server
        6.0
            AppPath=f:\iplanet\ias6\ias\APPS
            BasePath=f:\iplanet\ias6\ias
            ACL
            Admin
            CCS0
                CGI
                CONN
                ClusterName
                DAE
                DAE2
                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 Key value.



Key=

1XXX938X94-184XXX709X

5. Click OK.

6. Close the registry editor.

7. Restart iPlanet Application Server.

Changing the IP Address

When an iPlanet Application Server machine address changes, such as when the machine is moved, you must update the registry of that machine with the new address. If the machine participates in data synchronization, you must also update the registry of the other machines in the same cluster. Rather than locate every instance of the IP address in the registry and change each instance manually, you can use kregedit to update the entire registry with the new IP address.

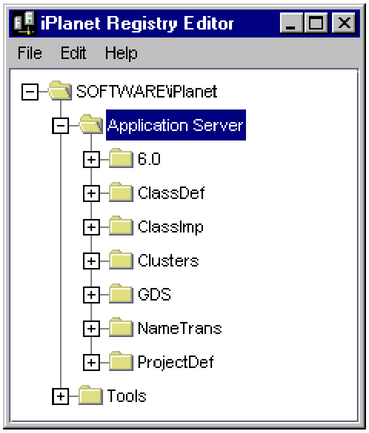
To change the IP address, perform the following steps:

1. Open the iPlanet Registry Editor by typing `kregedit` at the command line.

   (See "Using the iPlanet Registry Editor" on page 25.)

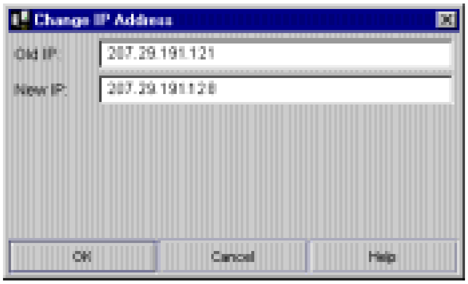
2. Open the following key:

   ```
   SOFTWARE\iPlanet\Application Server
   ```



3. From the Edit menu, choose Change IP Address.

   The following dialog box appears:



Old IP: 207.29.191.121

New IP: 207.29.191.128

4. Enter the old and new IP address.

5. Click OK to save your changes.

Part 2

Administering a Single iPlanet Application Server

Chapter 2

Monitoring Server Activity

This chapter describes the monitoring service provided by iPlanet Application Server Administrator. 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 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 interval instead of updating it continuously. You can specify this time interval in the Monitoring window. For information about setting the interval time, see "Changing a Process Data Plot" on page 40.

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.

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.
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.

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

Executive Server Process Attribute (KXS)	Description
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 the Load Balancing button. 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 the Rich Client, see the *Programmer'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.
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.

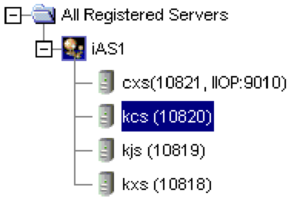
Table 2-2 Java, C++ and IIOP Bridge Server Monitoring Attributes *(Continued)*

Java/C++ and Bridge Server Processes (KJS, KCS and CXS) Monitoring Attributes	Description
Bytes received / interval	Number of new bytes received since the last polling.

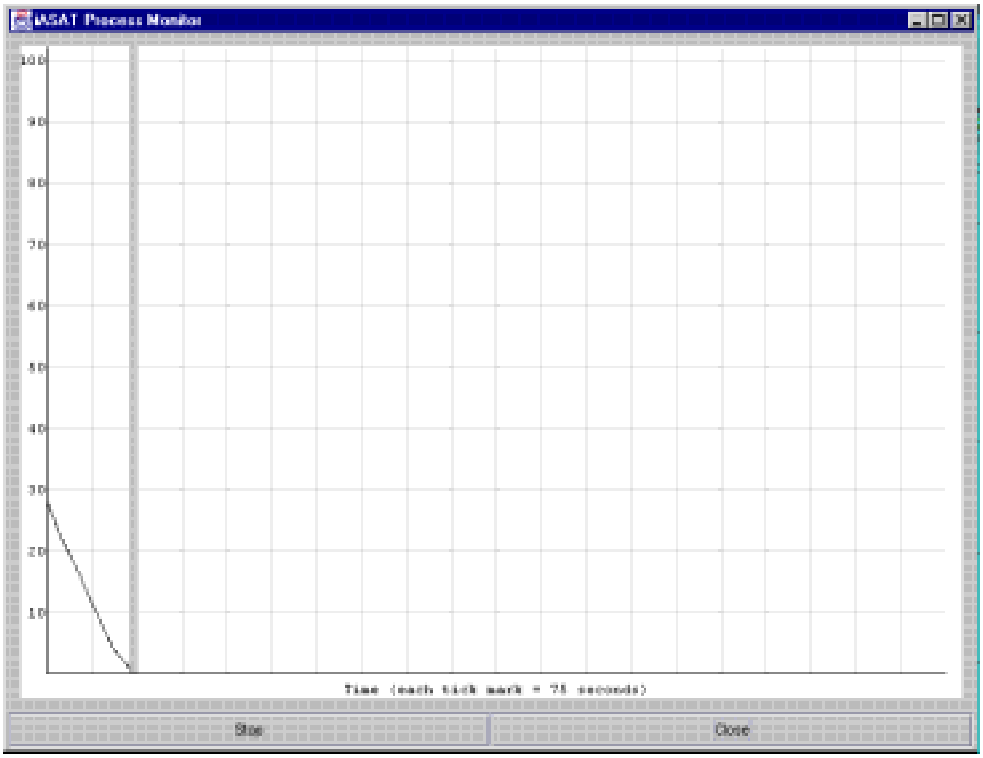
For each process, you can chart one or more attributes. You can also simultaneously chart the attributes of several application servers, if you have a multiple-server enterprise.

To monitor process attributes, perform the following steps:

1. On the iPlanet Application Server Administration Tool toolbar, click the Monitor button to open the Monitor window.
2. In the left pane of the Monitor window, click the process whose attributes you want to chart.

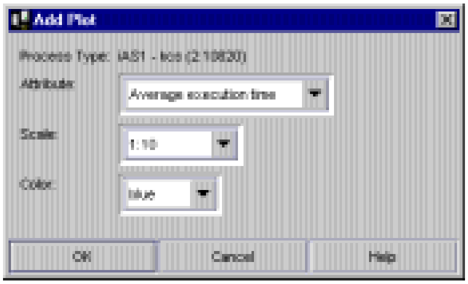


A separate monitoring panel pops out of the iPlanet Application Server Administration Tool:



3. In the right pane of the monitoring window in iPlanet Application Server Administration Tool, click the Add Plot button 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 from the Attribute drop-down list.

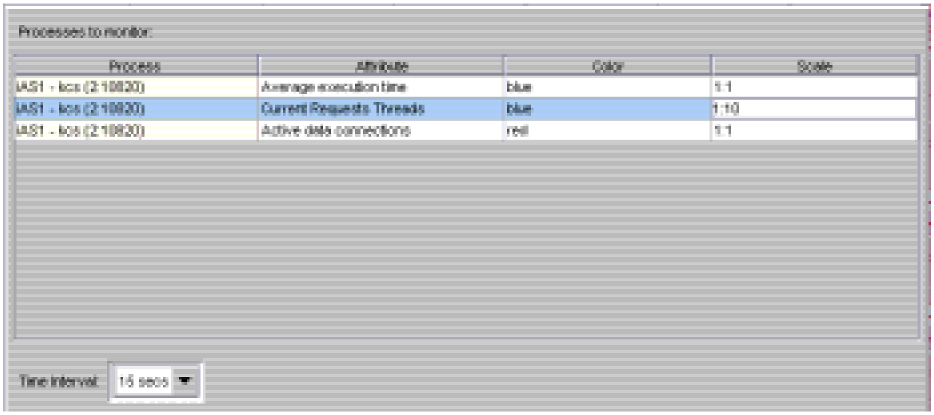
5. From the Scale drop-down list, choose the ratio (scale) at which to plot the attribute from the Scale drop-down list.

   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 from the Color drop-down list.

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)	Current Requests Threads	blue	1:10
iAS1 - kcs (2:10820)	Active data connections	red	1:1

Time interval: 15 secs

8. At the bottom of the Monitor window, specify how often you want to update the Monitor Plot window.

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

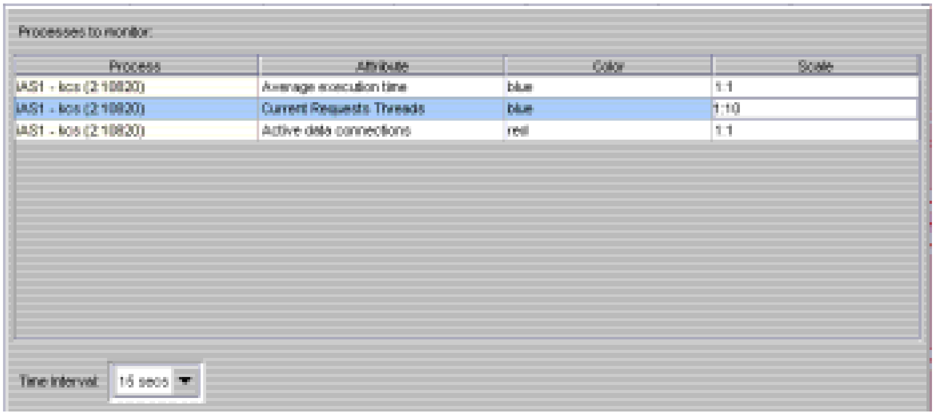
Logging Process Data to a File

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 iPlanet Application Server Administration Tool toolbar, click the Monitor button to open the Monitor window.

2. Click the process whose data you want to log as shown in the following figure:



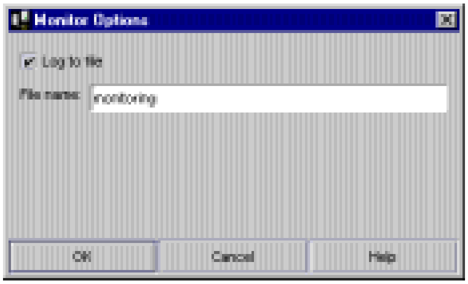
Processes to monitor:

Process	Attribute	Color	Scale
iAS1 - kcs (2:10820)	Average execution time	blue	1:1
iAS1 - kcs (2:10820)	Current Requests Threads	blue	1:10
iAS1 - kcs (2:10820)	Active data connections	red	1:1

Time interval: 15 secs

3. Click the Options button at the bottom of the window.

   The following dialog box appears:



Monitor Options

☑ Log to file

File name: monitoring

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 written.
6. Click OK.

Changing 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 iPlanet Application Server Administration Tool toolbar, click the Monitor button to open the Monitor window.
2. Locate a row where you want to change the plot.

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.

Removing 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 the Monitor button on the iPlanet Application Server Administration Tool toolbar to open the Monitor window.
2. Select the attribute for a process you want to remove.
3. Click the Remove Plot button.

   The attribute is removed from the Monitor window and is no longer plotted.

Receiving Event Notification

Event notification is useful when you cannot actively monitor an iPlanet Application Server machine. 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 via email when a critical situation arises by supplying the email address(es) of those you want to alert. In addition, you can specify a script that will run automatically when certain events occur.

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

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 process, 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.

Enable 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 Chapter 13, “Managing Distributed Data Synchronization.”

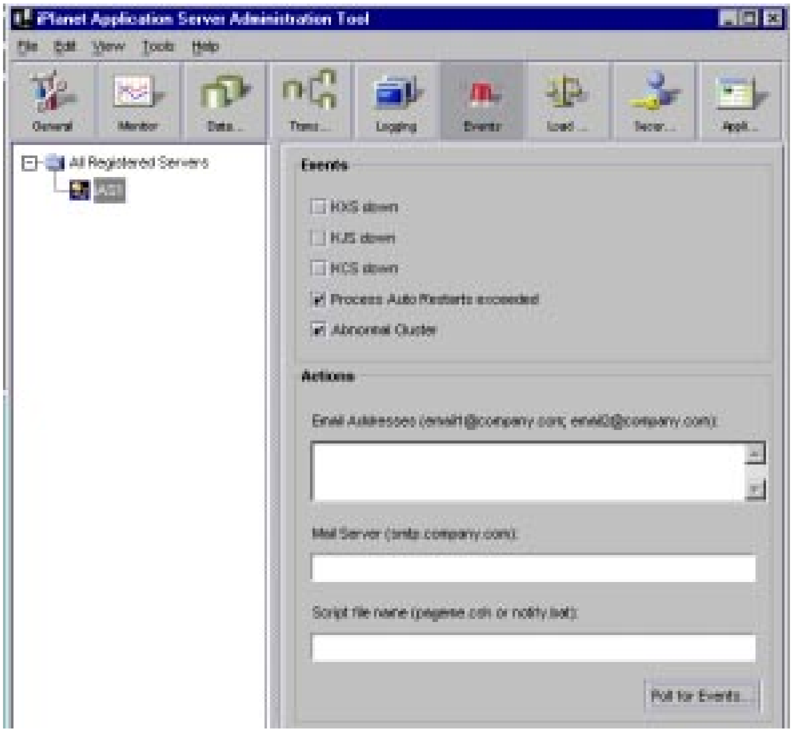
Configuring Email Notification for an Event

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

1. On the iPlanet Application Server Administration Tool toolbar, click the Events button 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:



iPlanet Application Server Administration Tool
File
Edit
View
Tools
Help
General
Monitor
Data...
Trans...
Logging
Events
Load...
Secur...
Appl...
All Registered Servers
Events
Actions
Email Addresses (email1@company.com; email2@company.com):
Mail Server (smtp.company.com):
Script file name (pageme.cdh or notify.bat):
Poll for Events...

☐ KXS down
☐ KJS down
☐ KCS down
☑ Process Auto Restarts exceeded
☑ Abnormal Cluster

4. In the Email Addresses field, specify the email address or addresses of the persons you want to send notification. Use the following format:

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

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 that when you click the Poll for Events button, 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 your application server.

Specifying 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 iPlanet Application Server Administration Tool toolbar, click the Events button 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, click the checkbox for the events 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 to the application server.

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.

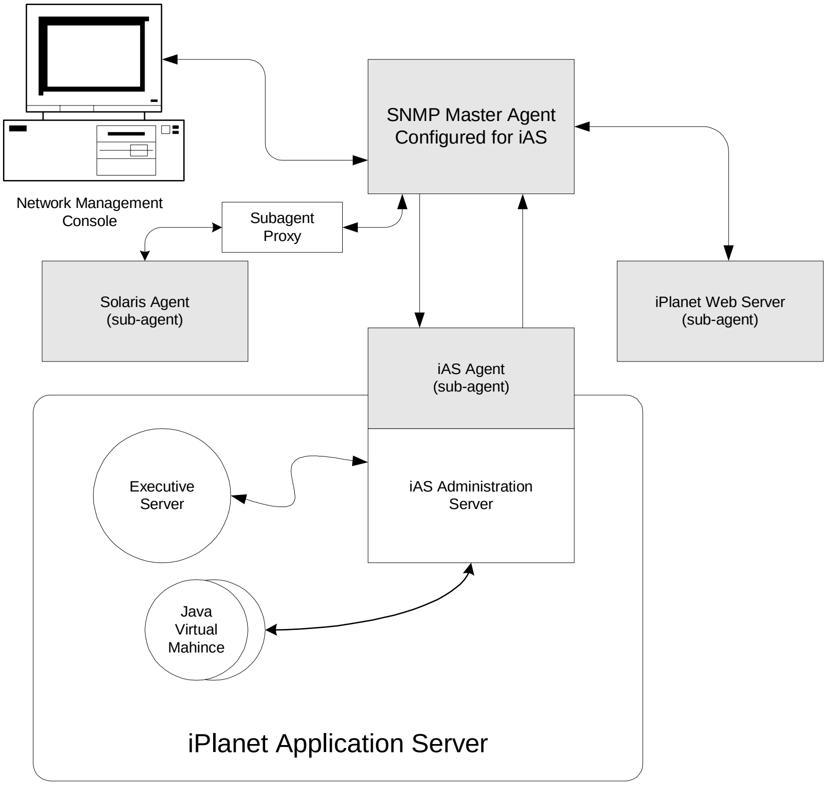

SNMP Master Agent Configured for iAS
Network Management Console
Subagent Proxy
Solaris Agent (sub-agent)
iPlanet Web Server (sub-agent)
iAS Agent (sub-agent)
Executive Server
iAS Administration Server
Java Virtual Mahince
iPlanet Application Server

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:

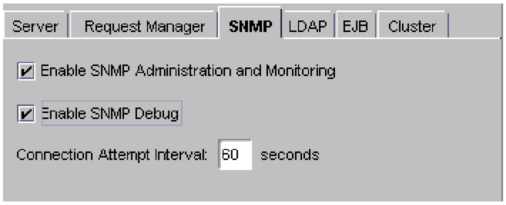
<iASInstallDir>/mail-instanceName/`log/default`

If the SNMP data collection process (`snmpcoll`), is not running, check the Administration Server Console to see whether the SNMP enable flag is on. For more information, see *Managing Servers with Netscape Console* on the following web site (`http://home.netscape.com/eng/server/console/`).

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

To enable data collection, perform the following steps:

1. Click the General button on the iPlanet Application Server Administration Tool toolbar to open the General window. Finally, click the SNMP tab.



☑ Enable SNMP Administration and Monitoring

☑ Enable SNMP Debug

Connection Attempt Interval: 60 seconds

2. Check 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.

3. Check the Enable SNMP Debug to log error messages if there is a problem with connecting to the master agent.

4. Specify the Connection Attempt interval.

   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.

Your configuration information is stored in Directory Server.

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.

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 read / write 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`.

  `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 in the following location:

`iASInstallDir/ias/snmp/gxnas.mib`

Additional iAS MIBs are located at:

`iASInstallDir/``plugins/snmp/`

Note that the additional MIBs are not required for iPlanet Application Server SNMP monitoring.

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.

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.

Starting the SNMP Master Agent

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

Verifying 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.

   `iASInstallDir``/ias/bin/KIVAes stop`

   Also make sure that all iPlanet Application Server processes (kas, kxs, kjs, kcs) are stopped. You can determine if any of them are running by using the UNIX ps command.

2. Verify that the LDAP server (slapd) is running using the UNIX ps command. If it is not running start slapd as follows:

   `iASInstallDir``/slapd-snickers/start-slapd`

   where “snickers” is the servername.

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

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

   where “snickers” is the servername.

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/S75snmpmagt 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.

```
<iASInstallDir>/ias/bin/KIVAes.sh start
```

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 DSync Logging Options

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. There 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.

Determining 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:

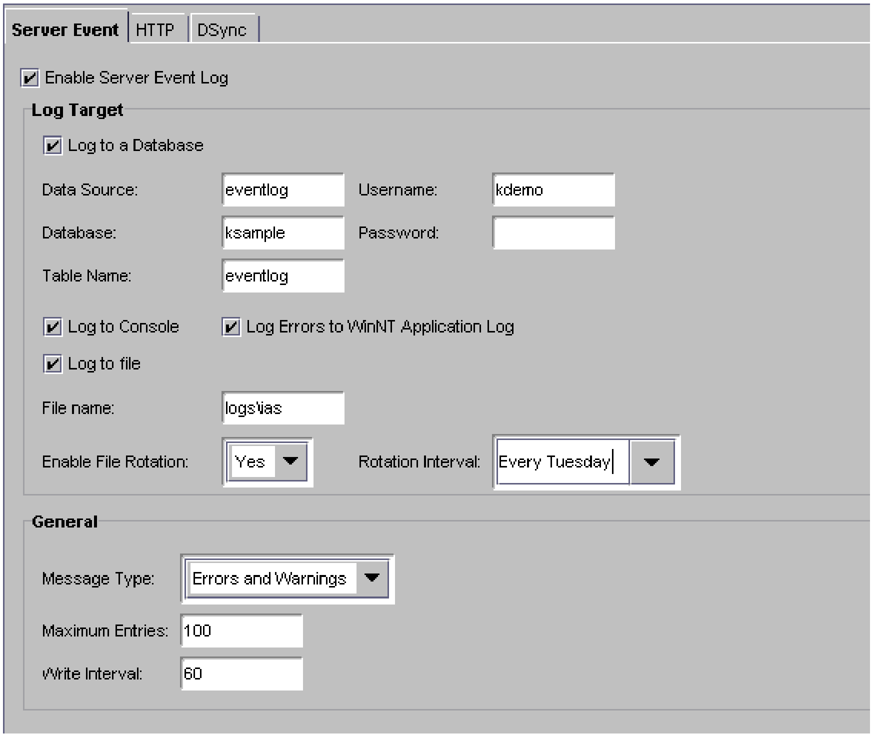
Table 4-1 Log Message Types

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 steps:

1. Click the Logging button on the iPlanet Application Server Administration Tool toolbar to open the Logging window.

2. Select the Enable Server Event Log 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: Every Tuesday

General

Message Type: Errors and Warnings

Maximum Entries: 100

Write Interval: 60

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

4. In the Maximum Entries text field, enter the maximum number of entries that can exist before data is written to the log.

5. In the Write Interval text 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 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 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	The 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. You can disable this feature by deselecting the Log to Console checkbox.

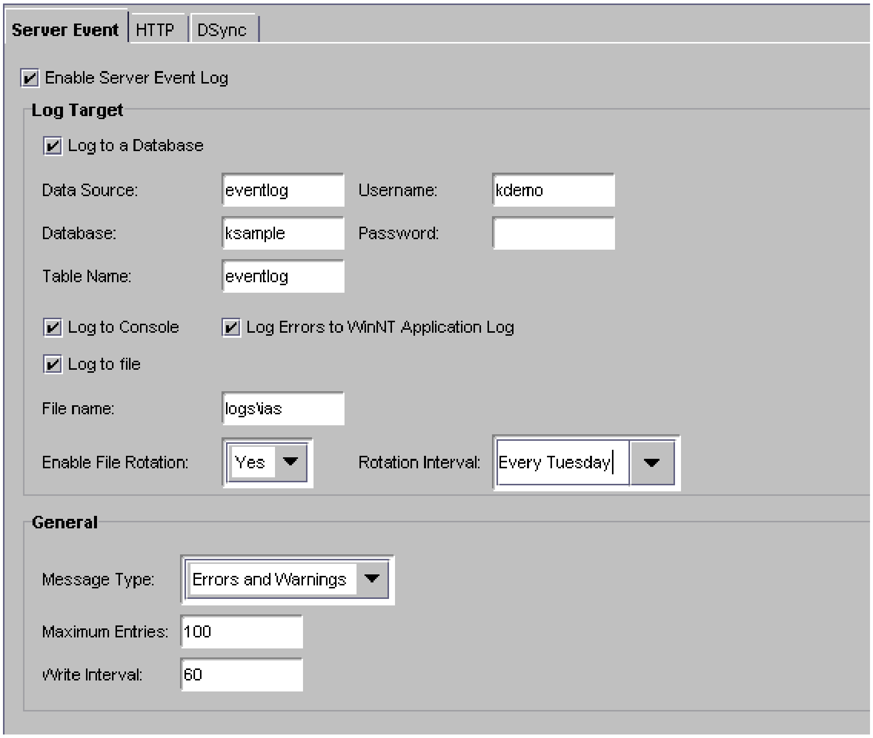
Table 4-2 Message Logging Destinations *(Continued)*

Log destination	Description	When to use
Application log	The default application log file. For Windows NT, this is viewable through the Event Viewer.	This is the default. Provides a more comprehensive record of the server and application error messages. Warning and information messages are not logged to the application log. 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 NT and Unix platforms, as long as those consoles are open and console logging enabled. On Windows NT, 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 steps:

1. Click the Logging button on the iPlanet Application Server Administration Tool toolbar to open the Logging window.
2. Select the Enable Server Event Log checkbox.



☑ 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

3. In the Log Target box, choose the type of logging to enable by clicking the desired checkboxes:

   - Log to a Database
   - Log to Windows NT Application Log (Errors Only)
   - Log to File

   You can disable console logging by deselecting the Log to Console checkbox.

   See "Logging to a Database" in the following section and "Logging to a File" on page 62 for more information.

   If you chose to log to a file, that file is created now. See "Rotating Log Files" on page 63 for information about managing log files.

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. This buffer optimizes the performance of the application server by limiting the use of resources to continually update a log. 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.

Logging to a Database

If you plan 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 a UNIX system, you can use supplied scripts that automatically set up the eventlog and httplog tables. The scripts are located in the directory `$GX_ROOTDIR/APPS/GXApp/Logging/db`, 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.

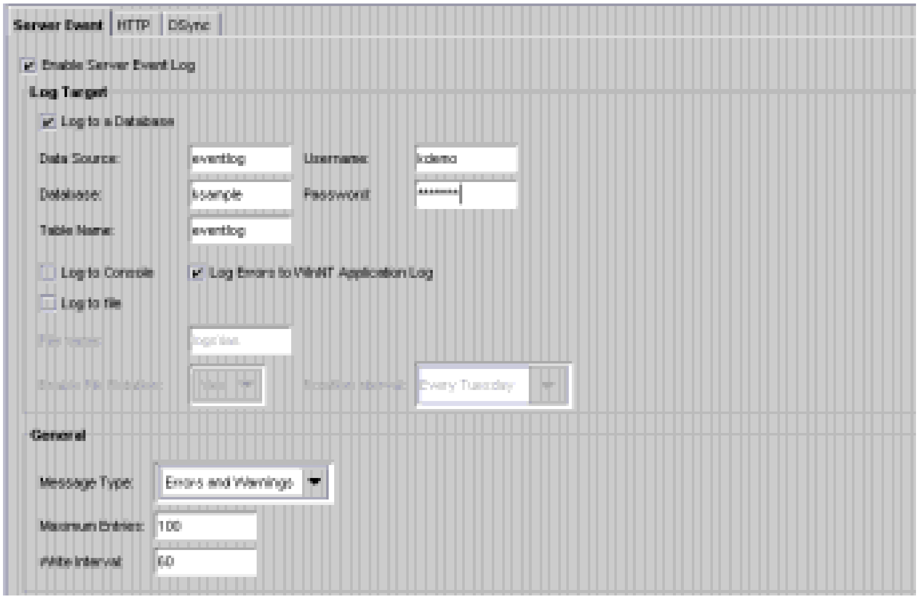
Table 4-3 Logging to a Database Table

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

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 database, perform following steps:

1. Click the Logging button on the iPlanet Application Server Administration Tool toolbar to open the Logging window.
2. Select the Enable Server Event Log 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

Rotation interval: Every Tuesday

General

Message Type: Errors and Warnings

Maximum Entries: 100

Write Interval: 60

3. In the Log Target box, click the Log to Database checkbox.

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

4. In the General box, from the Message Type drop-down box, select Errors, Errors and Warnings, or All Messages.
5. Click the Apply Changes button to save your changes to iPlanet Application Server Administration Tool.

Logging to a File

iPlanet Application Server Administration Tool's monitoring service allows you to log information about server activity to a file.

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

1. Click the Logging button on the iPlanet Application Server Administration Tool toolbar to open the Logging window.
2. Select the Enable Server Event Log checkbox.

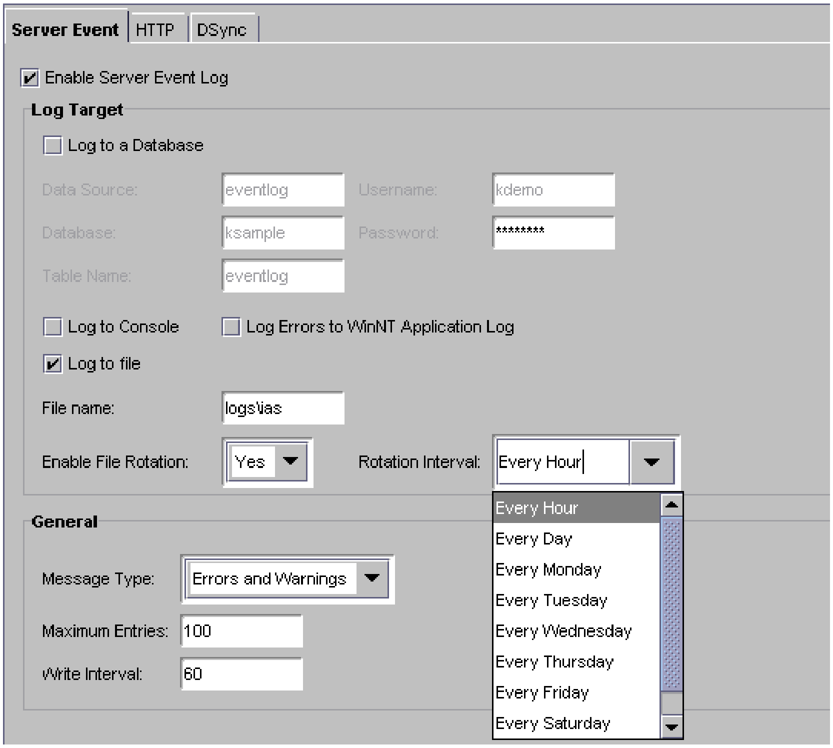
3. In the Log Target box, select the Log to File checkbox.
4. In the Log to File text field, enter the name of the log file.
5. In the General box, from the Message Type drop-down box, select Errors, Errors and Warnings, or All Messages.
6. Click Apply Changes to save your changes to iPlanet Application Server Administration Tool.

Rotating Log Files

If you choose to record server messages in an ASCII file, you can enable log file rotation to regulate when log files are rotated. Since log files are stamped with the time and date they are created, log file rotation helps organize log files into manageable units.

To configure log file rotation, perform the following steps:

1. Click the Logging button on the iPlanet Application Server Administration Tool toolbar to open the Logging window.
2. Select the Enable Server Event Log checkbox.



Server Event | HTTP | DSync

☑ 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. Click the Log to File checkbox.

4. In the Enable File Rotation drop-down box, choose Yes.

5. From the Rotation Interval drop-down box, select the interval at which log files are rotated or 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/*/*
   ```

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

7. Click Apply Changes to save your changes to the iPlanet Application Server Administration Tool.

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 machine where they are processed. By logging web server requests, you can track request patterns and other important request information.

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 a UNIX system, you can use supplied scripts that automatically set up the HTTP log and event log tables. See “Logging to a Database” on page 61 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

Table 4-4 HTTP Variables and Database Data Types

Default HTTP variables	Default database field name	Data type
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 must have a field name called `logtime` in the database table. The time the message is created is assigned by the logging service. The logging service maps that time to the `logtime` database field. You can rename all of the other database field names.

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. Even though this happens automatically during installation, there may be occasions when you must manually configure the web server.

Logging Web Server Requests

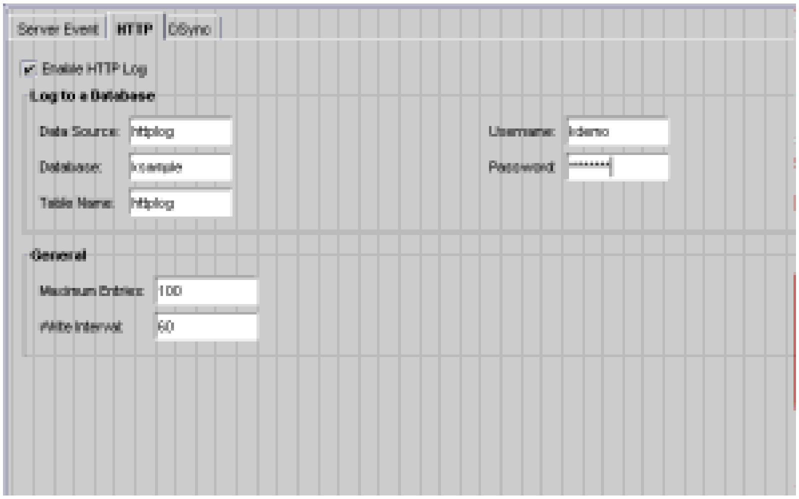
Before you can log web server requests, you must create a database table to hold the request messages. For more information about creating this table, see "How Web Requests Are Logged" on page 65.

To log web server requests, perform the following steps:

1. Click the Logging button on the iPlanet Application Server Administration Tool toolbar to open the Logging window.
2. From the left pane of the Logging window, select the application server responsible for logging web server requests.

   If you have more than one application server, you can specify one server to log web server requests.
3. In the right pane of the logging window, click the HTTP tab.

The following window appears:



Server Event | HTTP | DSync

☑ Enable HTTP Log

Log to a Database

Data Source: httplog

Database: ksample

Table Name: httplog

Username: kdemo

Password: ••••••••

General

Maximum Entries: 100

Write Interval: 60

4. Enter `httplog` in the Data Source field.
5. Enter 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. Enter the user name and passwords with which you connect to the database. Enter the maximum entries.

   This number represents the greatest number of entries that can exist before data is written to the log.
8. Enter the write interval.

   This number represents the amount of time that lapses before data is written to the log.
9. Click Apply Settings to save your changes to your application server.

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. The iPlanet Application Server Administration Tool provides for logging of DSync messages.

For more information about distributed data synchronization, see "About Distributed Data Synchronization" on page 195.

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
```

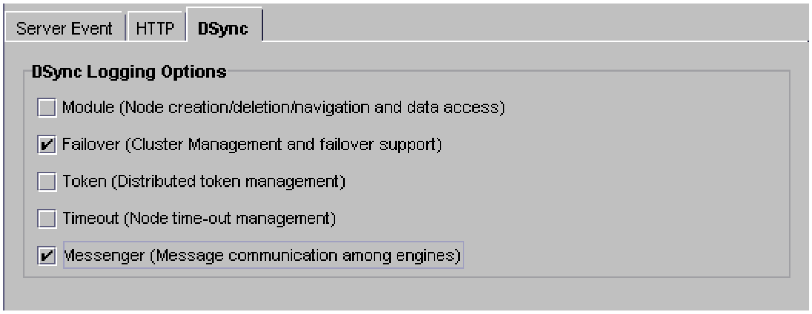
```
Entry[1]: ID=/dsync41test/U/4, expired 4 seconds ago
Entry[2]: ID=/dsync41test/W/4, expired 3 seconds ago
Entry[3]: ID=/dsync41test/V/4, expired 3 seconds ago
Entry[4]: ID=/dsync41test/X/4, expired 3 seconds ago
Entry[5]: ID=/dsync41test/D/5, expired 2 seconds ago
Entry[6]: ID=/dsync41test/Z/4, expired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

Logging DSync Messages

To log DSync debug messages, perform the following steps:

1. Click the Logging button on the iPlanet Application Server Administration Tool toolbar to open the Logging window.
2. From the left pane of the Logging window, select the 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 as follows:

- 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.

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

5. Optionally, 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" on page 68.

6. Optionally, 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" on page 69.

Chapter 5

Securing Applications

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

About Security

Implementing application security is a joint effort between the application developers and the server administrator: the application developers are responsible for determining what 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 the iPlanet Application Server Administration Tool.

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.

Limitations of This Document

This chapter does not explain Directory Server and iPlanet Console in great detail. Rather, it 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.

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 (LDAP) used with your instance of iPlanet Application Server. The information held in Directory Server is shared between all application servers when you have multiple servers supporting an application

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" on page 76.

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 username and password. For example, some applications authenicate a user via 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 *Programmer'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 *Programmer's Guide* 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.

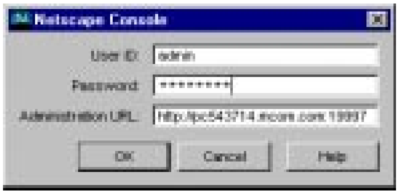
You must be a Directory Server administrator or a user with the necessary permissions to create a user.

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

1. From the Windows Start menu, under Programs, choose iPlanet Server Family, then iPlanet Console 4.0 to open iPlanet Console.

   For Unix, in the server root, enter `./startconsole`.

   The iPlanet Console login dialog box appears:

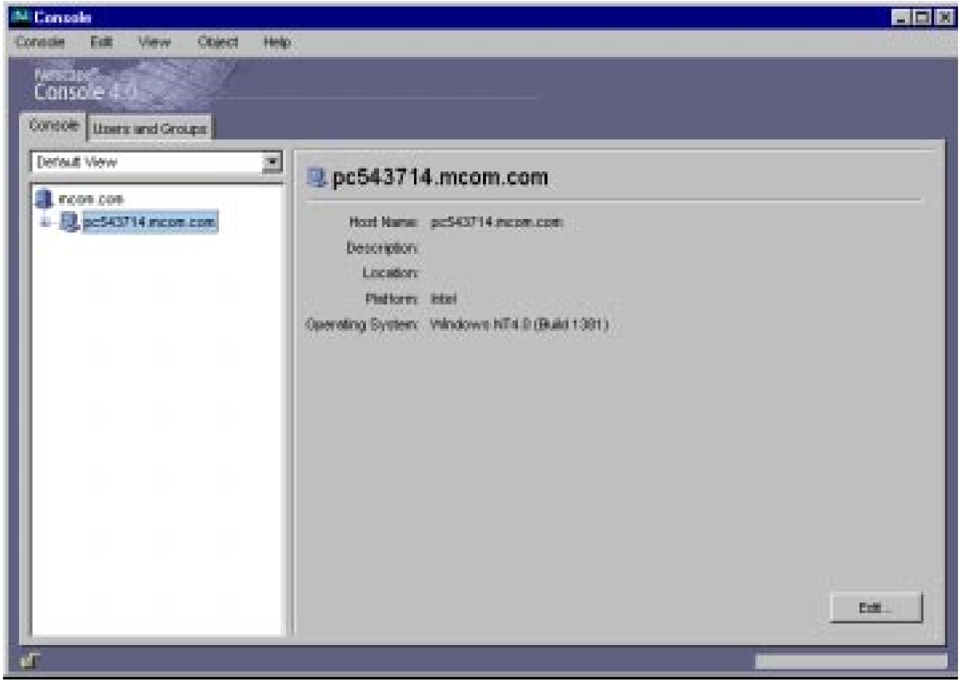
   User ID: admin

   Password: *********

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

2. Enter a valid user name and password and click OK.

   iPlanet Console’s main window appears:

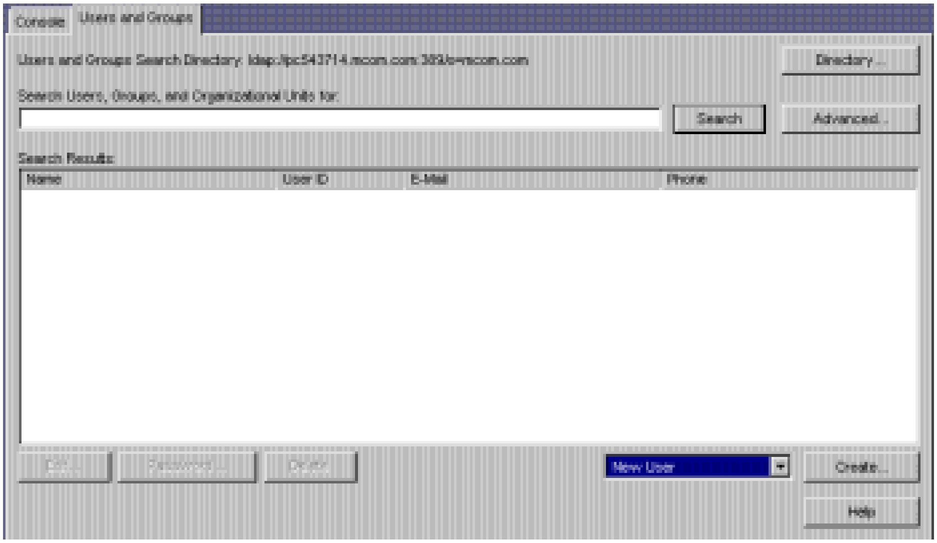


pc543714.mcom.com

Host Name: pc543714.mcom.com
Description:
Location:
Platform: Intel
Operating System: Windows NT4.0 (Build 1381)

3. Click the Users and Groups tab.

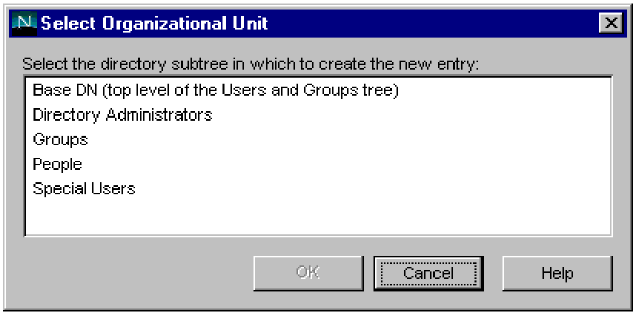
The following window appears:



Users and Groups Search Directory: ldap://pc543714.mcom.com:389/o=mcom.com

4. Use the drop-down list in the lower-right corner of the window to choose New User, then click Create.

   The Select Organizational Unit dialog box appears:

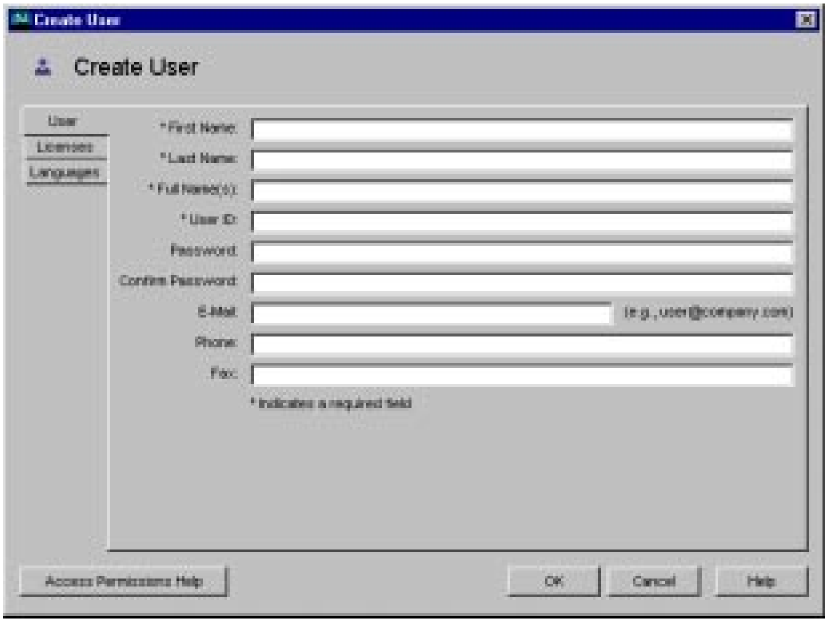


Select the directory subtree in which to create the new entry:

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

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

   The Create User window appears:

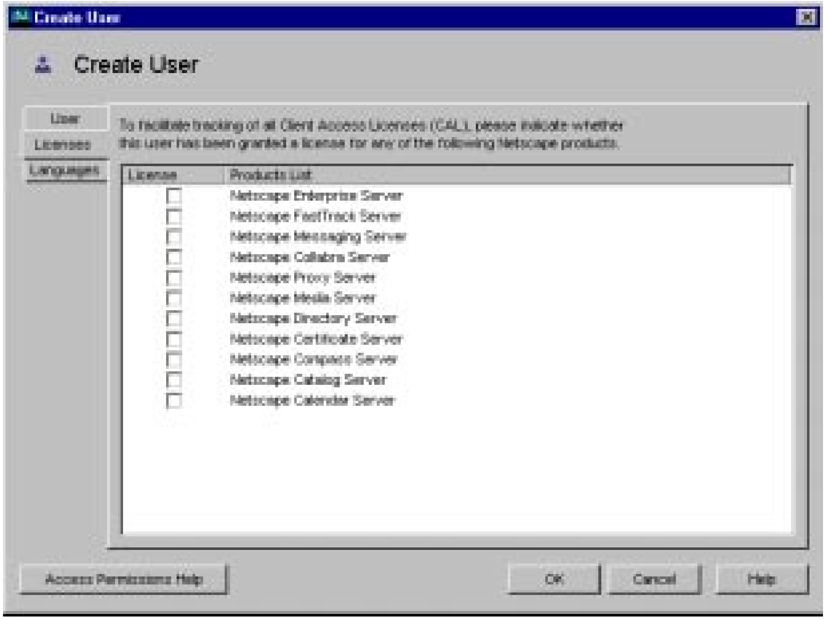


6. In the Create User window, enter user information.

- Full Name(s) 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 as necessary.
- 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.

7. Click the Licenses tab.

   The following window appears:



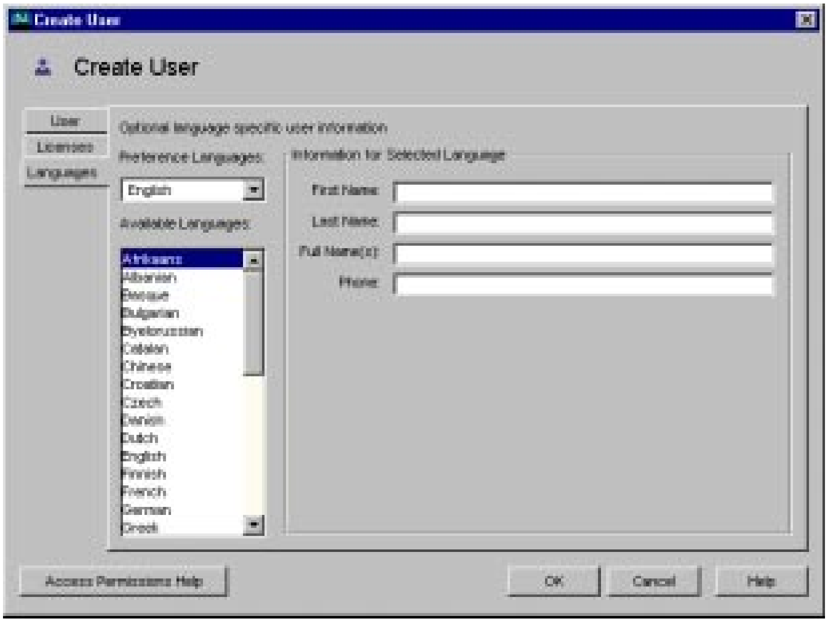
To facilitate tracking of all Client Access Licences (CAL), please indicate whether this user has been granted a license for any of the following Netscape products.

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

8. Select the servers this user is licensed to use, then click OK.

9. (Optional) Click the Languages tab.

   The following window appears:



Preference Languages: English

- ❍ 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.

Creating Group Entries Using iPlanet Console

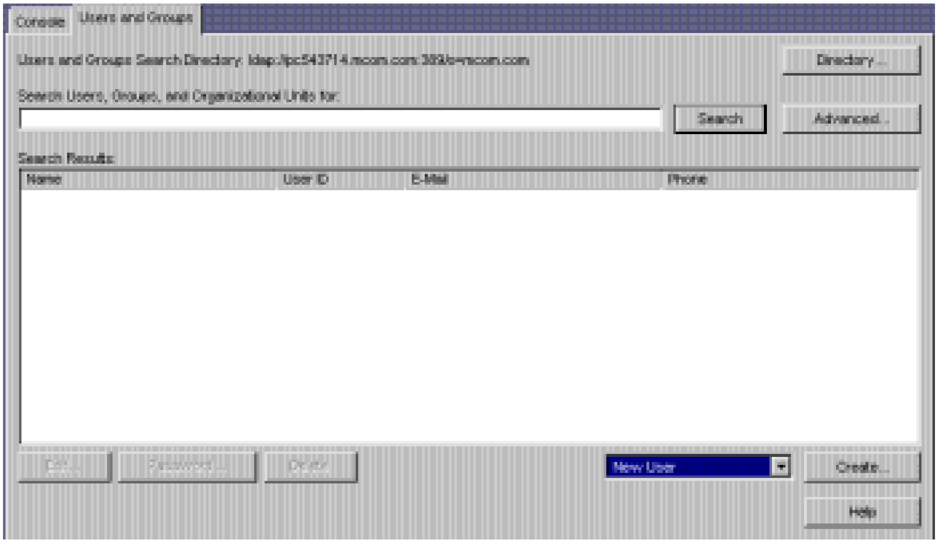
A group consists of all users who share a common attribute. For example, all users with DNs containing the attribute `ou=Sales` belong to the Sales group. Once you create a new group, you add users, or members, to it. You can use three types of groups in your directory: static, dynamic, and certificate 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 have the attribute `department=marketing` in their DN. None of those users are members of the Marketing group until you explicitly add each one to the 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://pc543714.mcom.com:389/o=mcom.com

Search Users, Groups, and Organizational Units for:

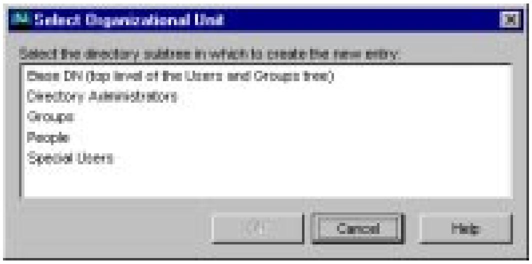
Search Results:

Name	User ID	E-Mail	Phone

New User

2. Use the drop-down list in the lower-right corner of the window to choose New Group, then click Create.

   The following dialog box appears:



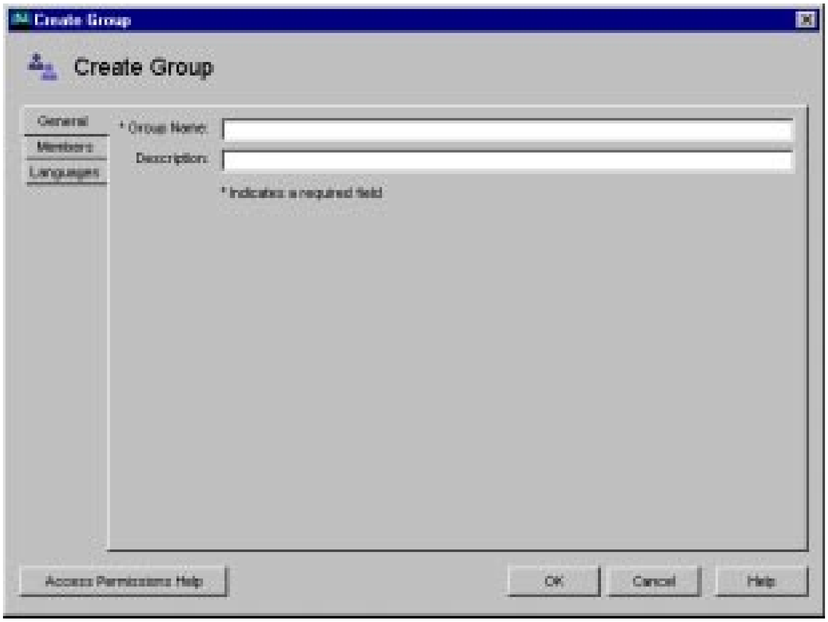
Select Organizational Unit

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

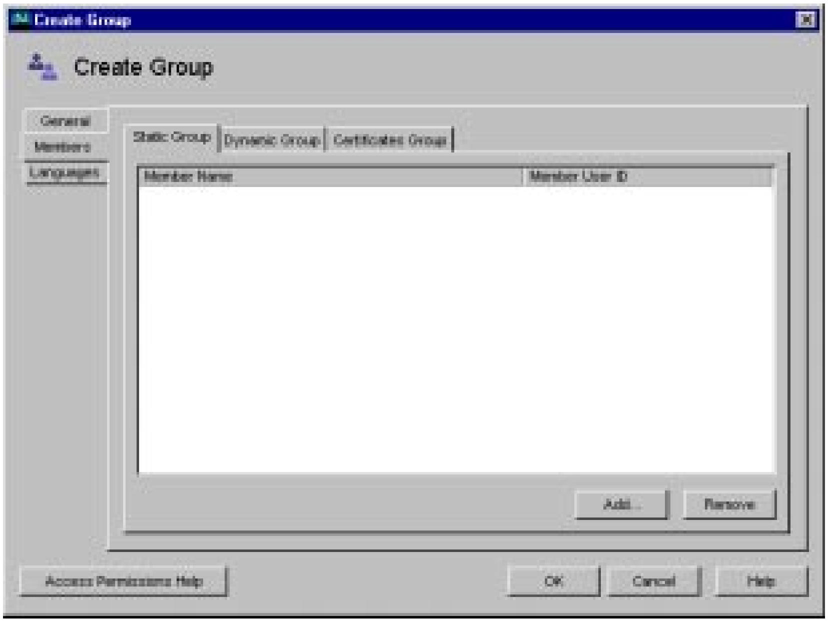
3. In the Select Organizational Unit window, select the directory subtree (ou) to which the group will belong, then click OK.

   The Create Group window appears:



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

   The following window appears:

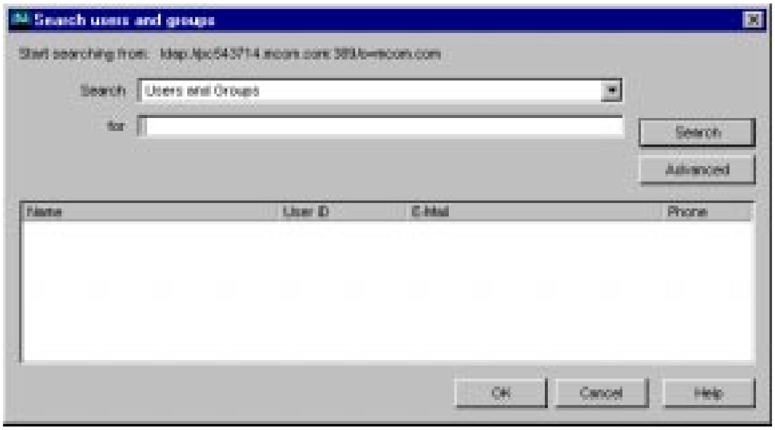


5. If you only want to create the group now and plan to add group members later, click OK and skip the rest of this procedure.

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

6. In the Members window, click Add or Edit as appropriate.

   The following dialog box appears:


Start searching from: ldap://pc543714.mcom.com:389/o=mcom.com

Search: Users and Groups

Use the Search dialog box to locate a user you want to add to the Members User ID list. 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. Then you can select operations from the menu bar to change the entry. The operations you perform apply to all in the Search list.

See iPlanet Console documentation for more information.

Using LDIF to Add Entries to Directory Server

You can add entries to Directory Server using LDIF or iPlanet Console. iPlanet Console is described "Using iPlanet Console to Add Entries to Directory Server" on page 76.

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 remaining three—`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 *Programmer's Guide* 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)” on page 87.
- For C++ Applications, authorization is set by permissions in access control lists. See “Setting Access Control List Authorization (for C++ Applications)” on page 89.

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 use iPlanet Application Server Administration Tool to manage roles of deployed applications. When managing 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.

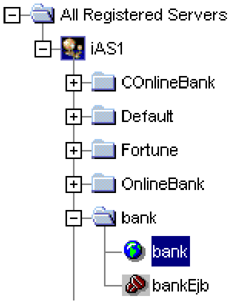
NOTE Roles for servlets and EJBs are created in the deployment descriptor XML files before deployment. See the online help that is provided with the Deployment Tool for more information.

To manage a role, perform the following steps:

1. On the iPlanet Application Server Administrator toolbar, click the Application button to open the Application window.

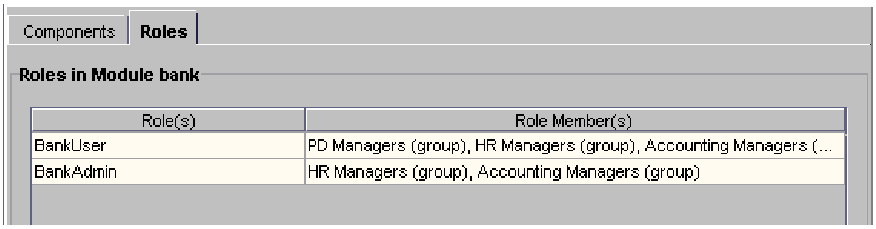
2. In the left pane, expand the iPlanet Application Server instance where the application is deployed.

3. Open the application folder and highlight a servlet or EJB icon.



All Registered Servers
- iAS1
  - COnlineBank
  - Default
  - Fortune
  - OnlineBank
  - bank
    - bank
    - bankEjb

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

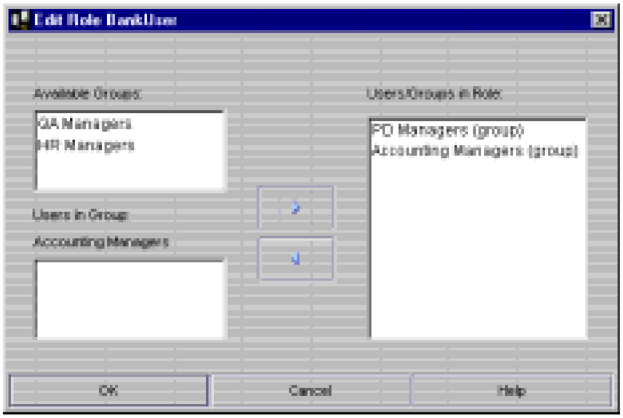


Roles in Module bank

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 the Edit Role button.

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 that 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

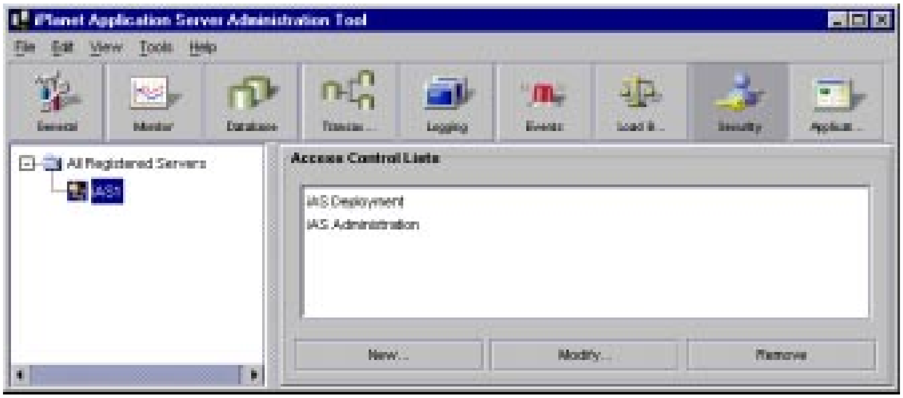
You use iPlanet Application Server Administration Tool to create and manage access control lists (ACLs). When creating an ACL, you can create groups to which users belong and add only groups to the ACL rather than adding individual users as members to the ACL. This is useful if you are using individual 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 iPlanet Application Server Administration Tool toolbar, click the Security button to open the Security window.

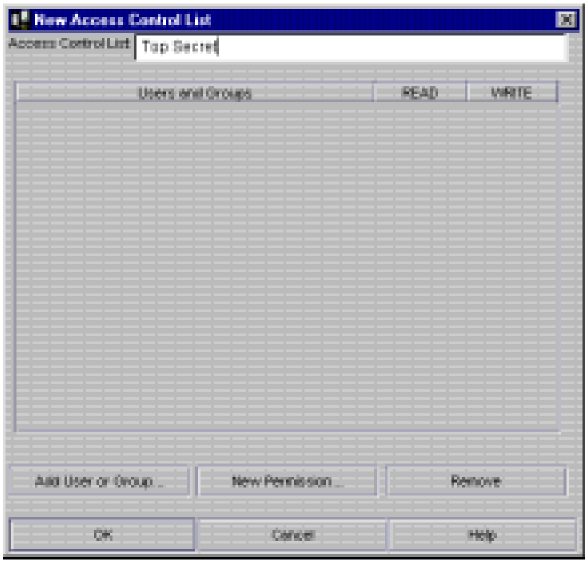
   The following window appears:



iAS Deployment
iAS Administration

2. Click the New button located at the bottom of the window.

   The New Access Control List dialog box appears.



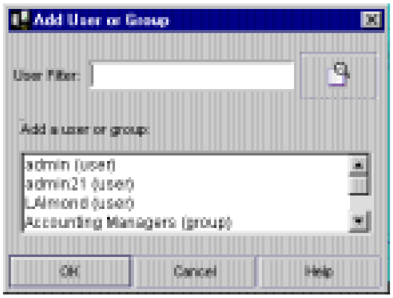
Access Control List: Top Secret

3. In the Access Control List 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 the Add User or Group button at the bottom of the dialog box.

   The Add User or Group dialog box appears.



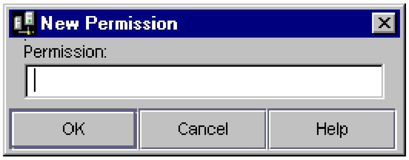
5. Select the users and / or groups you want to add to the ACL.

   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 User Filter button. 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. Enter the 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:

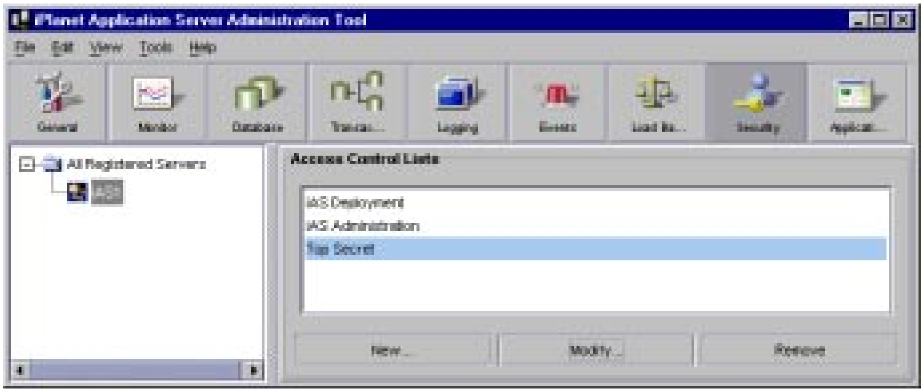
- add groups
- create new permissions
- edit permissions

You can also remove groups from the system.

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

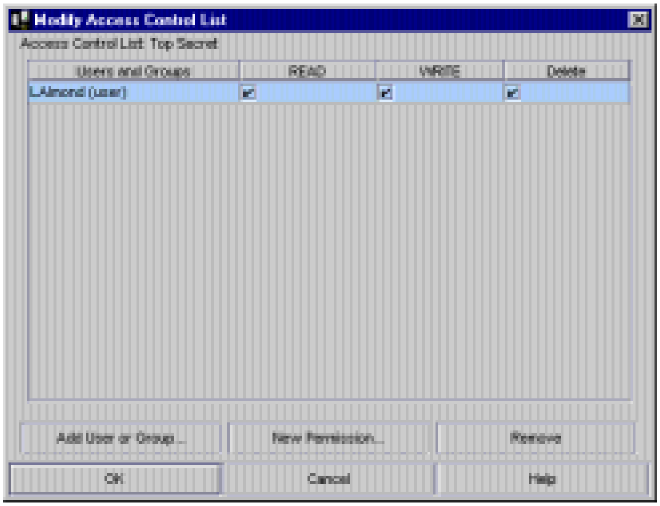
1. On the iPlanet Application Server Administration Tool toolbar, click the Security button to open the Security window.

   The following window appears:



2. Highlight the Access Control List that you want to modify.
3. Click the Modify button located at the bottom of the window.

   The Modify Access Control List dialog box appears.

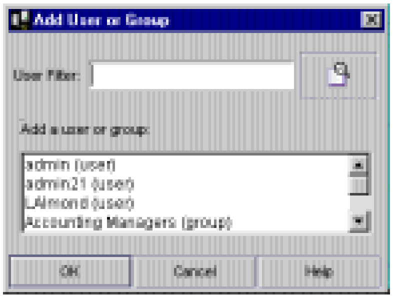


Access Control List: Top Secret

Users and Groups	READ	WRITE	Delete
LAlmond (user)	☑	☑	☑

4. To add a new user or group, click Add User or Group.

   The Add User or Group dialog box appears.



User Filter:

Add a user or group:

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

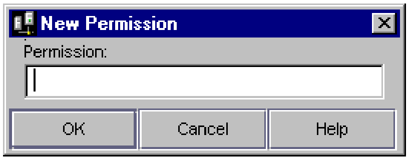
5. Select the group or groups you want to add to the ACL.

   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 User Filter button. 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 create a new permission, click New Permission.

   The New Permission dialog box appears.



8. To edit the permissions of a group, select or deselect the appropriate permissions for that group.

9. To remove a group, select that group and click Remove.

Chapter 6

Increasing Fault Tolerance and Server Resources

This chapter describes increasing iPlanet Application Server resources, which can increase application performance.

The following topics are included in this chapter:

- About Adding and Tuning Server Processes
- Adjusting the Number of Threads for a Process
- Setting Options of the Administrative Server
- Implementing a Multi-Process, Single-Threaded Environment
- Configuring Directory Server Failover

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

When planning how to increase server resources, you must take into account the resources of the iPlanet Application Server machine. For instance, if the machine is not capable of handling additional processes, you can negatively affect the performance of an application by increasing the number of processes running on that machine. 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.

About Adding and Tuning Server Processes

You can add a Java Server (KJS) or C++ Server (KCS) process to increase fault tolerance. By having one or two additional processes, an application is more likely to respond to users' requests. If one process fails, for instance, the second or third process can take its place, decreasing the amount of time an application is unavailable. This is particularly useful for applications that have known problems that 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 an iPlanet Application Server Server using RMI/IIOP. When a request originates from a Rich Client, it is sent to iPlanet Application Server by way of an Bridge process. This allows Rich Clients to communicate directly to application components on iPlanet Application Server.

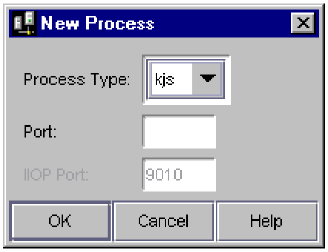
Adding and Tuning Java and C++ Processes

You can add more KJS processes for Java applications and add more KCS processes for C++ applications. It is usually not necessary to add more than two processes for each type of application, Java and C++. If an application cannot run on one or two processes, there are most likely errors in the code that are causing the processes to fail. Those errors should be addressed by the application developer.

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

1. On the iPlanet Application Server Administration Tool toolbar, click the General button to open the General window.
2. In the left pane of the General window, select the iPlanet Application Server machine where you want to add the KJS process.
3. From the File menu, choose New, then Process.

   The Add Process dialog box appears.



New Process

Process Type: kjs

Port:

IIOP Port: 9010

4. In the Process drop-down box, 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 steps:
    a. Click the process in the left pane of the General window.
    b. In the right pane of the window, set the Default Minimum and Default Maximum Threads to 1.
8. Click the Apply Changes button to save your changes.

Adding 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 via a web path where requests originate at a Web Browser and then are 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 *Programmer's Guide*.

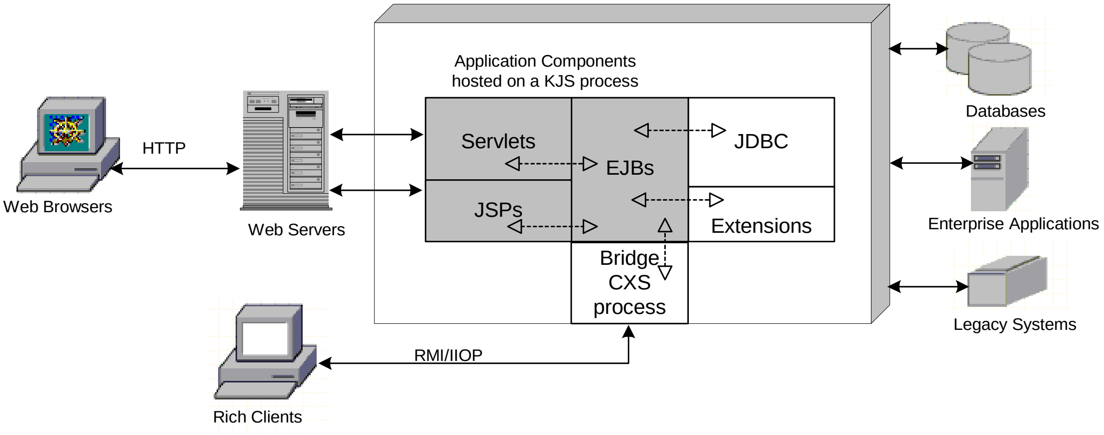
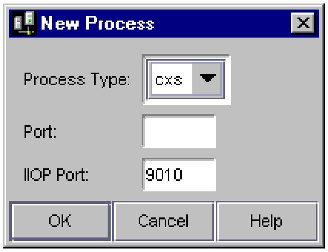


Figure 6-1 iPlanet Application Server Communication Architecture

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

1. On the iPlanet Application Server Administration Tool toolbar, click the General button to open the General window.
2. In the left pane of the General window, select the iPlanet Application Server machine where you want to add the CXS process.
3. From the File menu, choose New, then Process.

   The Add Process dialog box appears.



Process Type: cxs

Port:

IIOP Port: 9010

4. In 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. if this process is to be used in a single-threaded environment, perform the following steps:
   a. Click the process in the left pane of the General window.
   b. In the right pane of the window, set the Default Minimum and Default Maximum Threads to 1.
9. Click the Apply Changes button to save your changes.

Adjusting the Number of Threads for a Process

Request threads handle users' requests for application components. When iPlanet Application Server receives a request, the application server 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. When the resource is free, the thread allows 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 those 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 specify on an as-needed basis.

Increasing the number of threads available to a process to allow that process to respond to more application requests simultaneously. Threads can be added to a process at the process level, or globally at the iPlanet Application Server level.

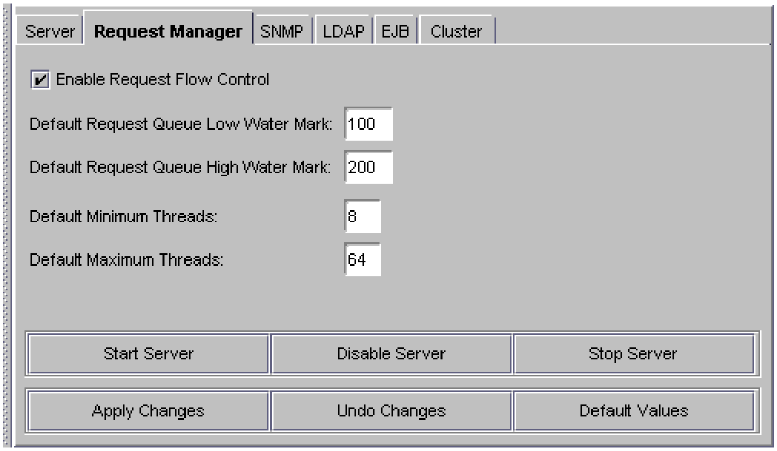
By default, each process uses the threads assigned to iPlanet Application Server. For example, if iPlanet Application Server uses a minimum of 8 threads and a maximum of 64 threads, each individual process uses a minimum of 8 threads and a maximum of 64 threads.

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

1. On the iPlanet Application Server Administration Tool toolbar, click the General button and then the Request Manager tab.

2. In the left pane of the General window, select the server whose number of threads you want to adjust.

3. In the Default Minimum Threads text box, enter the minimum number of threads available for each process on the selected iPlanet Application Server machine.



☑ 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

4. In the Default Maximum Threads text box, enter the maximum number of threads available for each process on the selected iPlanet Application Server machine.

5. Click Apply Changes to save your changes.

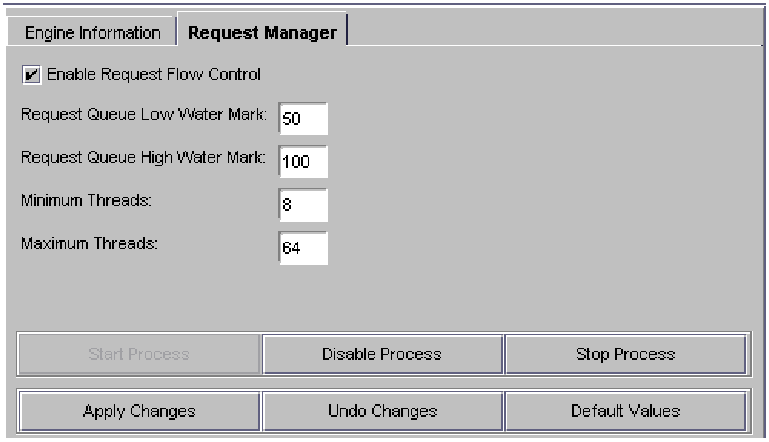
You can also customize the usage of threads for a process. Once you do this, however, the number you set at the process level overrides the number you set globally at the iPlanet Application Server level.

To adjust the number of threads available for a process, perform the following steps:

1. On the iPlanet Application Server Administration Tool toolbar, click the General button to open the General window.

2. In the left pane of the General window, select the process whose number of threads you want to adjust.

3. In the Request Manager box, enter the minimum number of threads available for that process.



☑ Enable Request Flow Control

Request Queue Low Water Mark: 50

Request Queue High Water Mark: 100

Minimum Threads: 8

Maximum Threads: 64

4. Enter the maximum number of threads available for that process.

   These settings override the default settings set at the server level.

5. Click Apply Changes to save your changes.

Specifying the Number of Requests for the Queue

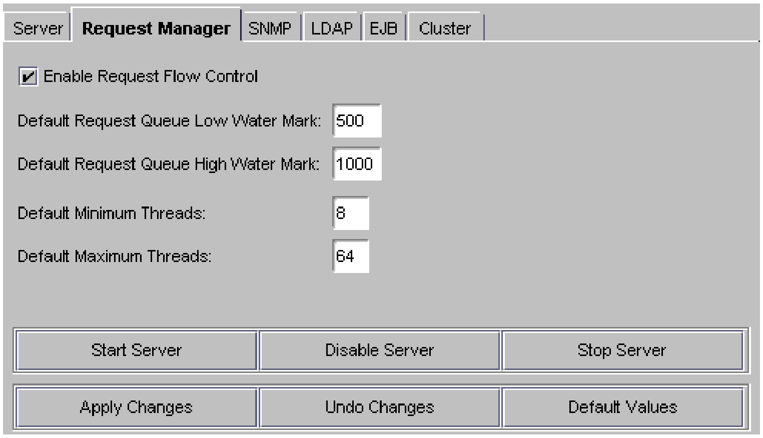
The web connector plug-in routes users requests to applications residing on iPlanet Application Server to the Executive process (KXS). These requests are logged to the request queue in the Executive process. You can control the maximum number of threads the web connector plug-in will use. This prevents the request queue from receiving more requests than it can process. To control the flow of requests, you can set the maximum number of requests that are logged to the request queue. The maximum number is called the “high watermark.” When the request queue reaches this number no new requests will be accept and the web-server will return an error page for additional iPlanet Application Server requests. You can also set the number of requests in the queue in which logging will resume. This number is called the “low watermark.”

To control the flow of requests on the server level, perform the following steps:

1. On the iPlanet Application Server Administration Tool toolbar, click the General button to open the General window.

2. In the left pane of the General window, select the server in which you can to control request flow.

3. Click 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

4. In the Request Queue Low Water Mark text box enter the number of requests 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.

5. In the Request Queue High Water Mark text box enter 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.

6. Click Apply Changes to save your changes.

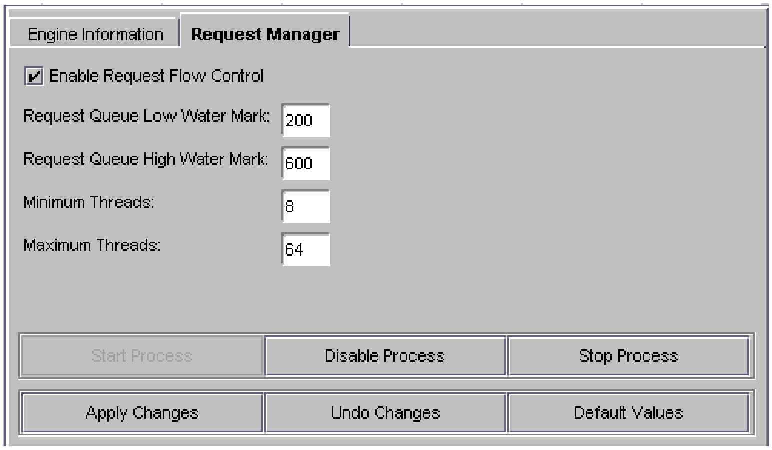
You can also customize the request flow for a process. Once you do this, however, the number you had set globally at the iPlanet Application Server level is overridden by the flow control numbers you set at the process level.

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

1. On the iPlanet Application Server Administration Tool toolbar, click the General button to open the General window.

2. In the left pane of the General window, select the process whose request flow you want to specify.

3. 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

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

5. 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.

6. Click Apply Changes to save your changes.

Setting Options of the Administrative Server

The Administrative Server is the administrative process within iPlanet Application Server through which administrative tasks are processed.

There are several options that you can set for the administrative server that will increase fault tolerance and server resources. This can increase the performance of applications running on a server and attempt to reduce the likelihood of application downtime. The following are the options you can set:

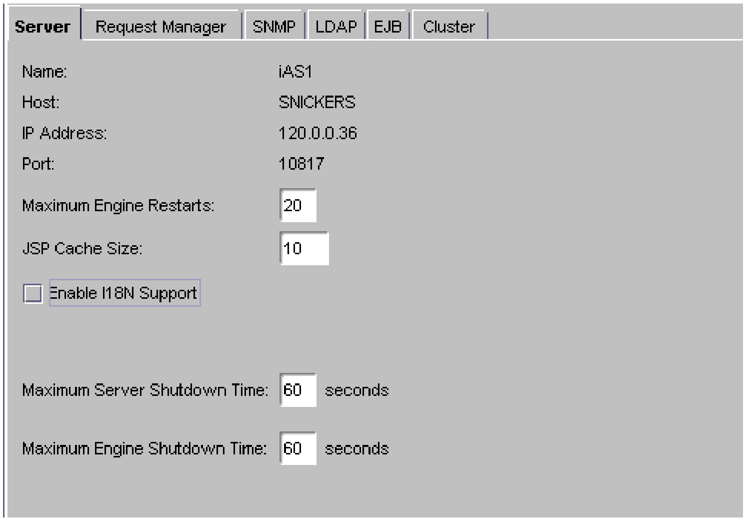
- Maximum Engine Restarts
- JavaServer Pages (JSP) Caching
- Maximum Server and Engine Shutdown Time
- Internationalization Support

Adjusting the Restart Option of the Administrative Server

Adjust the restart option of the Administrative Server to increase or decrease the number of times the Administrative Server attempts to restart an Executive Server (KXS), Java Server (KJS), C++ Server (KCS) or Corba Executive Server (CXS) that has failed. This option increases fault tolerance and application availability by attempting to ensure that all processes are running.

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

1. On the iPlanet Application Server Administration Tool toolbar, click the General button to open the General window.
2. In the left pane of the General window, select the iPlanet Application Server machine whose Administrative Server restart option you want to adjust.
3. On the Server tab, enter the new restart value in the Maximum Number of 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

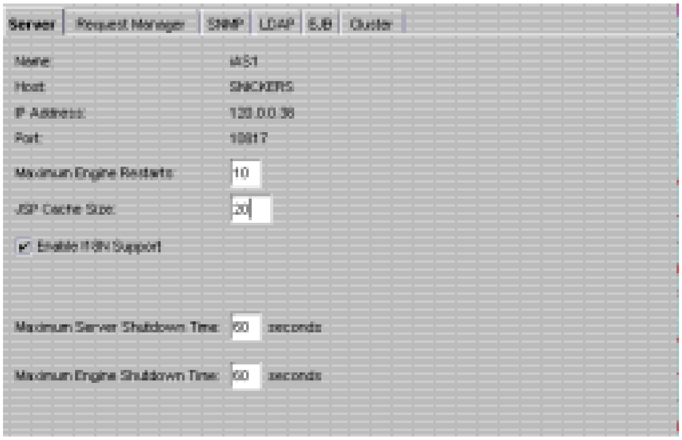
4. Click Apply Changes to save your changes.

Setting JSP Caching

You can set a JSP caching value of the Administrative Server to specify the number of JSP pages that are cached by each KJS engine on an iPlanet Application Server instance. Caching is set on a per-page basis. Caching JSPs optimizes application response time.

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

1. On the iPlanet Application Server Administration Tool toolbar, click the General button to open the General window.
2. In the left pane of the General window, select the iPlanet Application Server machine for which you want to set the JSP caching value.
3. On the Server tab, enter the JSP Cache Size in the text field. The cache size is set on a per-page basis.



Name: iAS1
Host: SNICKERS
IP Address: 120.0.0.38
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

4. Click Apply Changes to save your changes.

Enabling Internationalization

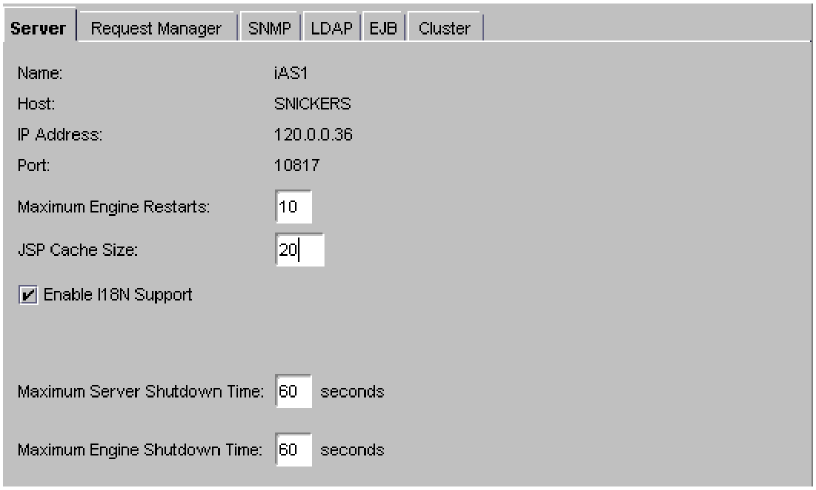
You can enable the capability of iPlanet Application Server to support applications of multiple locales.

To enable internationalization, perform the following steps:

1. On the iPlanet Application Server Administration Tool toolbar, click the General button to open the General window.

2. In the left pane of the General window, select the iPlanet Application Server machine for which you want to enable internationalization.

3. On the Server tab, check the I18N Support 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

4. Click Apply Changes to save your changes.

5. You must stop and restart the server for you changes to take affect:

   iASInstallDir/ias/bin/KIVAes.sh stop

   iASInstallDir/ias/bin/KIVAes.sh start

Setting Shutdown Time

You can set a shutdown value of the Administrative Server for both iPlanet Application Server and engine processes. For example, if you set a 60 seconds engine shutdown time, any application tasks currently being processed by the engines are allowed to complete within 60 seconds and no new requests are accepted. The ability to specify 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 iPlanet Application Server Administration Tool toolbar, click the General button to open the General window.

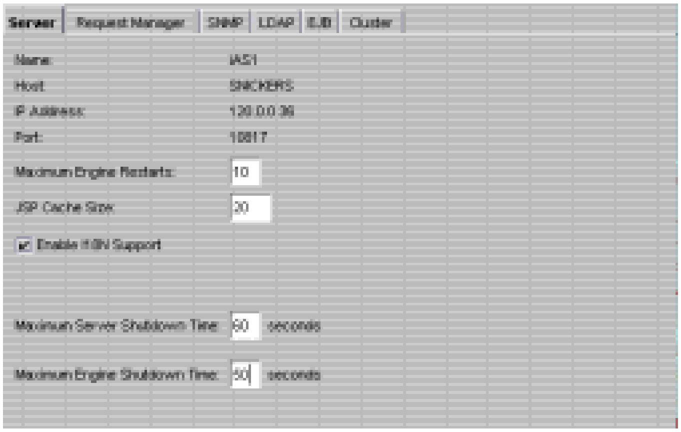
2. In the left pane of the General window, select the iPlanet Application Server machine whose shutdown time you want to specify.

3. On the Server tab, enter a Maximum Server Shutdown Time.

   The Maximum Server Shutdown Time is the maximum time 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 iPlanet Application Server will wait for any individual engine to shut down. After this time, the engine will be killed, and then the next engine 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 a multi-process, single-threaded environment allows each process to accept only one request at a time. This is useful when you are integrating third-party utilities. Running third-party utilities in the iPlanet Application Server multi-threaded request environment can cause errors beyond the control of the application server, including thread safety issues. To work around this type of problem and still allow the iPlanet Application Server to scale, you can implement a multi-process, single-threaded environment.

For example, if a third-party utility runs within the KJS process, but this utility is not thread safe, you can adjust the request threads of the KJS to 1 and eliminate the utilitiy's safety issues. However, this creates a request backlog as requests wait for the KJS to process a single request at a time. To alleviate that problem, you can run multiple KJS processes, all running 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 KJS or KCS processes.

   See "About Adding and Tuning Server Processes" on page 96.

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

   See "Adjusting the Number of Threads for a Process" on page 99.

Configuring Directory Server Failover

The Directory Server connected to your iPlanet Application Server machine 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 machines that share a single Directory Server. To protect this globally shared information, you must configure a second Directory Server to act as a backup if the primary server fails.

Before adding a backup Directory Server to your Directory Server cluster, you must replicate the iPlanet Application Server subtree of the primary Directory Server using supplier initiated replication (SIR). SIR is a replication configuration where servers containing master copies of directory trees and subtrees replicate directory data to servers containing replicated directory trees and subtrees.

The two copies of the iPlanet Application Server subtree must always be in sync with each other.

The iPlanet Application Server subtree is

```
cn=Global, cn=iasconfig, cn=iAScluster, o=iPlanetRoot
```

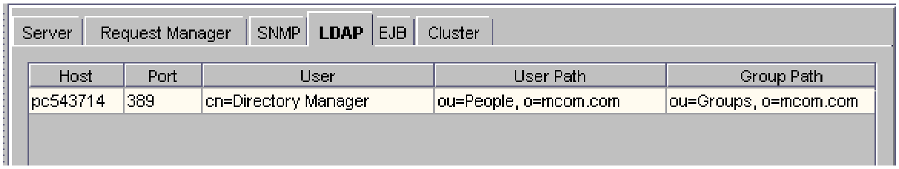
where `iasconfig` is specified during installation.

For details and replication procedures, see "Managing Replication," a chapter in *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
```

Now add a backup Directory Server using the iPlanet Application Server Administration Tool by performing the following steps:

1. On the iPlanet Application Server Administration Tool toolbar, click the General button to open the General window.
2. In the General window, click the LDAP tab to display the following screen:

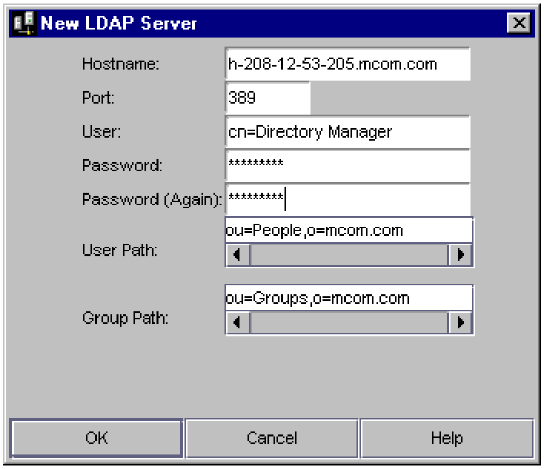


Host	Port	User	User Path	Group Path
pc543714	389	cn=Directory Manager	ou=People, o=mcom.com	ou=Groups, o=mcom.com

   Each Directory Server associated with your iPlanet Application Server machine appears in the window.

3. To add a secondary Directory Server, click the Add button.

   The following dialog box appears:



New LDAP Server

Hostname:	h-208-12-53-205.mcom.com
Port:	389
User:	cn=Directory Manager
Password:	*********
Password (Again):	*********
User Path:	ou=People,o=mcom.com
Group Path:	ou=Groups,o=mcom.com

4. Enter the new server's information.

5. Click OK.

To remove a Directory Server, click Remove.

You must always have at least one Directory Server configured to work with iPlanet Application Server.

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
- Configuring the Web Connector 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 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 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 functions:

Table 7-1 Configurable Web Connector 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 for Web Server Logging” on page 115
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” on page 117
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” on page 118
The plug-in port number	Reconfigure the port number used by the plug-in.	“Changing the Web Connector Port Number” on page 119
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” on page 119

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 will run as a CGI script. If you perform the entire procedure, the web connector 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 (iWS), perform the following steps:

1. Enable CGI, if it is not already enabled:

   a. From the Start menu, go to the iPlanet program group and choose Administer iPlanet 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 box and click Manage.

   d. On the main menu bar across the top of the page, click Programs.

   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 NT:

   drive letter`:\Netscape\Server4\docs\cgi-bin`

   For iPlanet Web Server 4.1, Unix:

   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 NT:

   drive letter`:\Netscape\Server4\https-`*machinename*`\config`

   For iPlanet Web Server 4.1, Unix:

   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 NT:

```
Init fn="load-modules"
   funcs=ias_name_trans,gxrequest,gxlog,gxinit,gxredirect,
   gxhtmlrequest shlib="path to iAS bin dir/example:
   gxnsapi351.dll"
```

```
Init fn="gxinit"
```

- Unix:

```
Init fn="load-modules"
   funcs=ias_name_trans,gxrequest,gxlog,gxinit,gxredirect,
   gxhtmlrequest shlib="gxnsapi30.so"
```

```
Init fn="gxinit"
```

Specify the following for `shlib`, iPlanet Enterprise Web Server 4.1:

- Windows NT:

  iASInstallDir`\bin\gxnsapi351.dll`

- Unix:

  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 NT:

drive letter`:\iPlanet\SuiteSpot\https-`*machinename*`\conf_bk`

For Unix:

iPlanet install directory/`https-`*machinename/*`conf_bk`

6. **Unix 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.

Reconfiguring 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 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.

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 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\INPUTISAPI
```

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.

Chapter 8

Administering Database Connectivity

iPlanet Application Server applications are able to access a database, or several databases, 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

About Data Access Drivers

iPlanet Application Server applications often require database access. Database access is achieved through a data access driver, which is 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

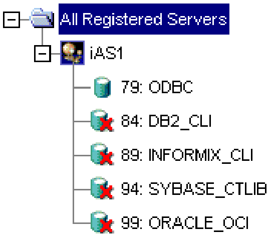
Make sure that data access drivers are installed before installing an instance of iPlanet Application Server. This way, iPlanet Application Server can automatically configure the drivers.

Configuring Data Access Drivers

When you open the Database window of iPlanet Application Server Administration Tool, 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:

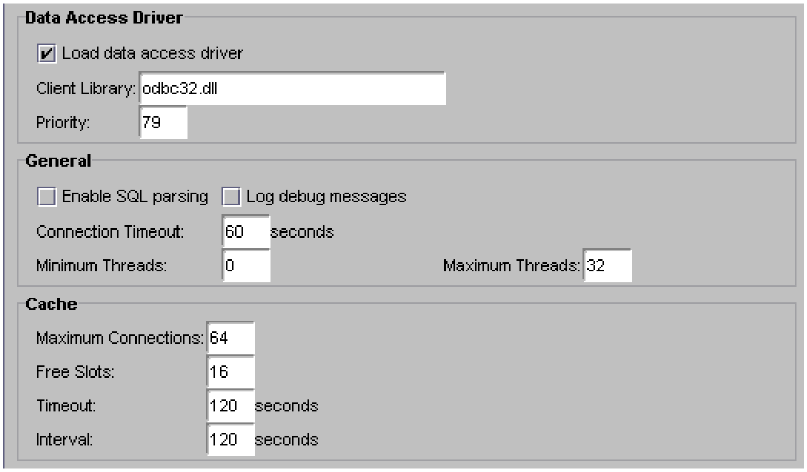
1. From the iPlanet Application Server Administration Tool toolbar, click the Database button to open the Database window.
2. In the left pane of the Database window, click the driver you want to configure.



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, click Load Data Access Driver.

   Information about the data access driver appears in the Database window.



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.

   Changes are not applied until you restart the server.

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:

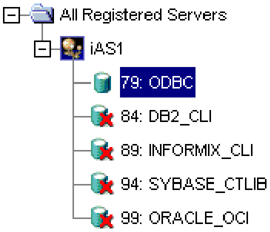
- connection
- threads
- result set buffer
- database cache

Setting Connection Parameters

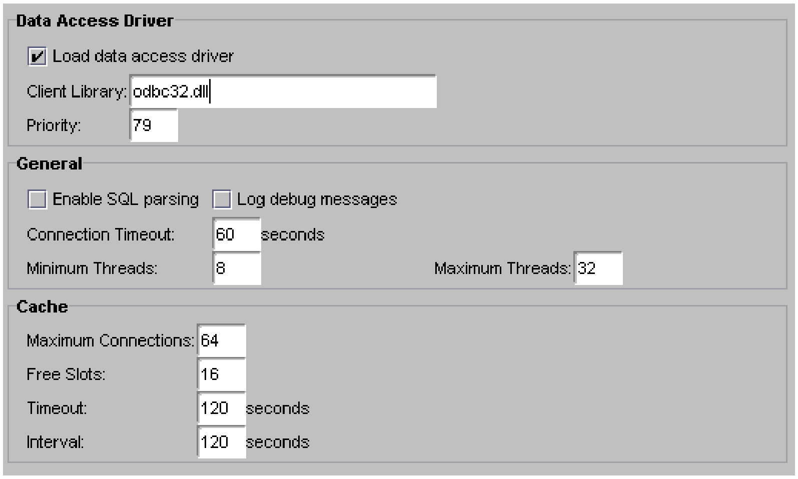
You can set the length of time iPlanet Application Server attempts to make a database connection. These parameters optimize the performance of the iPlanet Application Server machine by keeping the server from wasting resources. For example, because iPlanet Application Server waits for open database connections 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 iPlanet Application Server Administration Tool toolbar, click the Database button 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.



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.

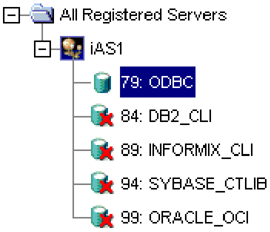
Setting Thread Parameters

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. Keep in mind that 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.

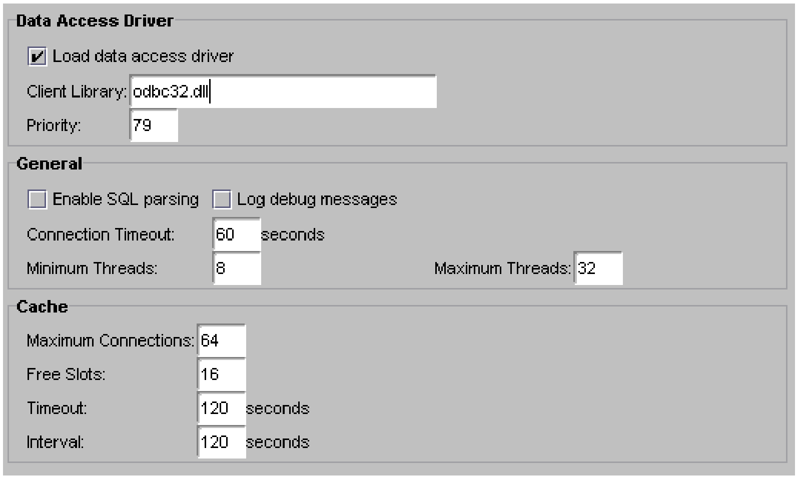
To set the thread parameters, perform the following steps:

1. From the iPlanet Application Server Administration Tool toolbar, click the Database button 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.



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 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.

Setting 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 finished, 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:

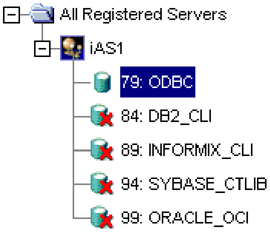
- the maximum number of connections allowed in the cache
- the number of slots held solely for free connections
- the timeout limit, in seconds, for free connections
- the interval, in seconds, at which the cache cleaner thread removes timed-out free connections

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.

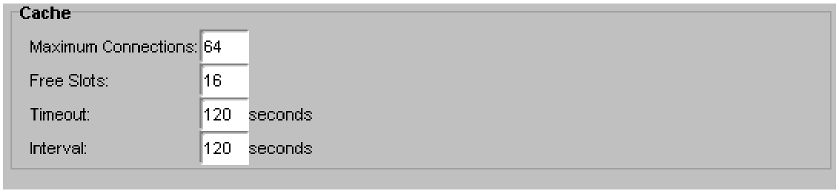
To set database cache parameters, perform the following steps:

1. From the iPlanet Application Server Administration Tool toolbar, click the Database button 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 seconds

Interval: 120 seconds

4. Click Apply Settings to save the changes to iPlanet Application Server.

Chapter 9

Administering Transactions

This chapter describes the tasks and conceptual information necessary for administering transactions using the iPlanet Application Server Administration Tool.

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 iPlanet Application Server Administration Tool.

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.

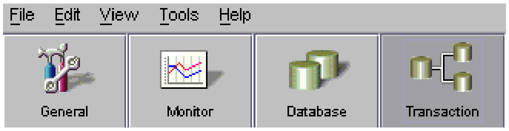
The following table lists the registry entries to which iPlanet Application Server refers along with their default values:

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

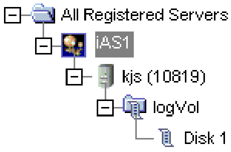
Administering Distributed Transactions in the Transaction Window

You can administer transactions using the Transaction window of iPlanet Application Server Administration Tool. To access the Transaction window from the iPlanet Application Server Administration Tool toolbar, click the Transaction button as shown in the following illustration:



About the Transaction Window

The left pane of the Transaction window displays a tree of nodes as shown in the following illustration:



All Registered Servers
iAS1
kjs (10819)
logVol
Disk 1

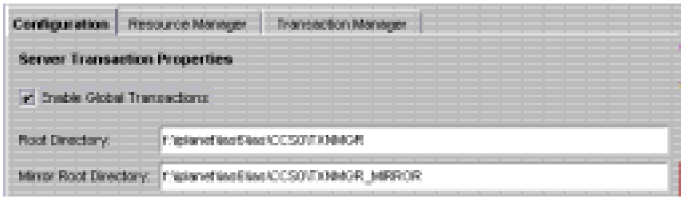
The top level of the tree lists which servers are registered with iPlanet Application Server Administration Tool. The second level, below each registered server name, displays one or more process nodes. These nodes indicate which processes are running on each registered server. Only Java Server (KJS) 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” on page 132 for more information about physical volumes.

When you click a physical volume node, the right pane of the transactions window displays the page size, or size of a page used in the transaction manager, the total size of the physical volume, and the amount of unused disk space in 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:



Server Transaction Properties

☑ Enable Global Transactions

Root Directory: f:\iplanet\ias6\ias\CCS0\TXNMGR

Mirror Root Directory: f:\iplanet\ias6\ias\CCS0\TXNMGR_MIRROR

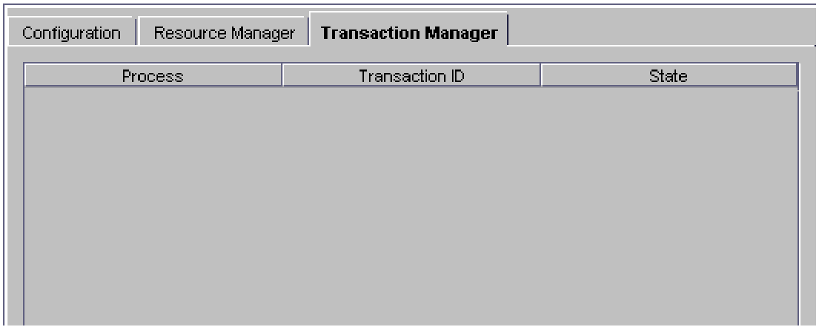
You can set the transaction mode. When global transactions are enabled, transactions can span across multiple heterogeneous databases and processes. When you clear the Enable Global Transactions checkbox, local transactions are enabled. Local transactions are limited to a single database / process but 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 following window appears:



Process	Transaction ID	State

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 the Update button 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 the Details button.

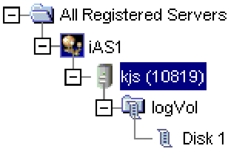
The Transactions Detail dialog box appears.

In the text box, Originator indicates where the selected transaction originates. The Participants box indicates where the transaction is currently running.

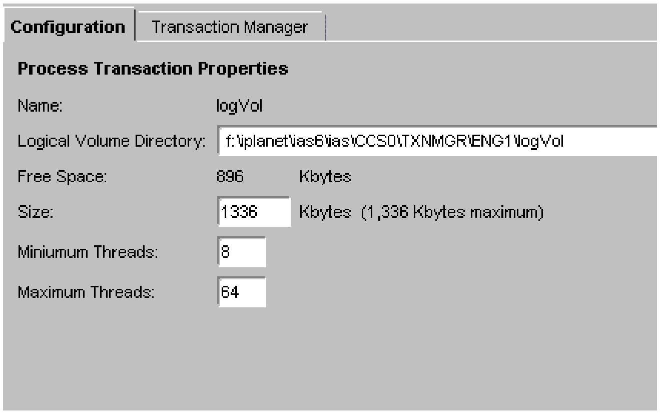
You can force the transaction into a state by clicking the appropriate button (Abort, Force Abort, Force Commit, Force Finish).

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)

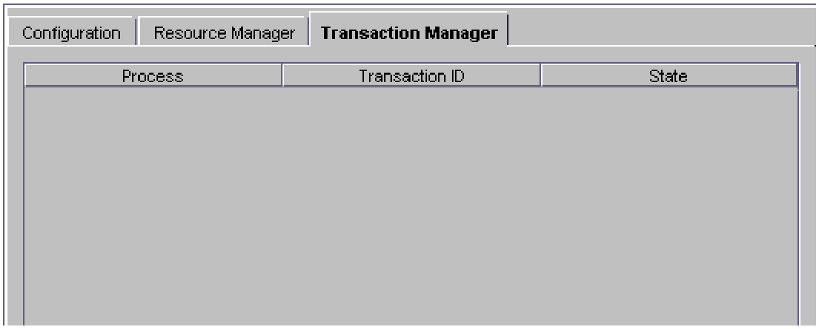
Miniumum Threads: 8

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:



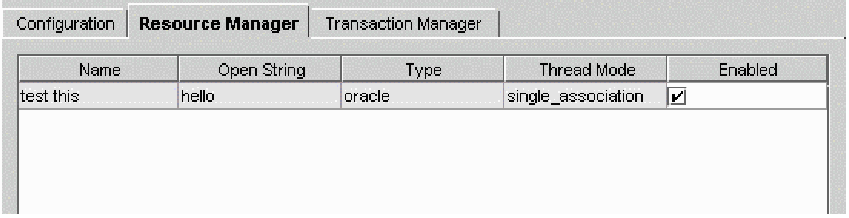
Process	Transaction ID	State

The transaction ID and state appear. See "Configuring Transactions per Server" on page 134 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:



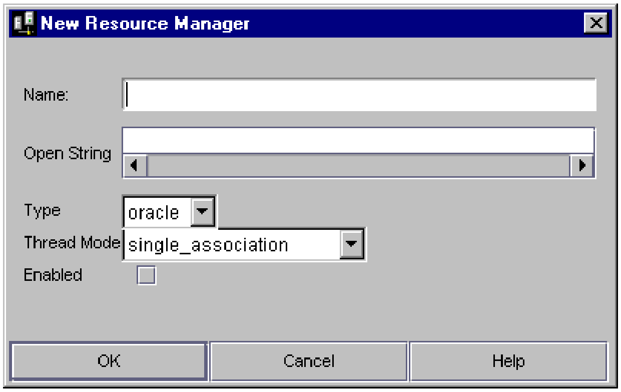
Name	Open String	Type	Thread Mode	Enabled
test this	hello	oracle	single_association	☑

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 that you must restart the server before changes to your resource manager configuration take effect.

Adding and Editing Resource Managers

To add or edit Resource Managers, perform the following steps:

1. Click the Add or Modify buttons to add or edit a resource manager. The following dialog box appears:



New Resource Manager

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

iasadmin Command	Function	Input parameter	Output parameter
`abort transaction`	Abort a server transaction.	`DWORD tid`	
`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`	

Table 9-2 Commands Executable from the Command Line *(Continued)*

iasadmin Command	Function	Input parameter	Output parameter
`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}`
`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}`

Table 9-2 Commands Executable from the Command Line *(Continued)*

iasadmin Command	Function	Input parameter	Output parameter
`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`	

The following table lists commands you can use in addition to those provided by `iasadmin`. As shown in the following example, these commands are not preceded by `iasadmin` at the command line.

```
%set server
```

Table 9-3 Additional Commands Executable From the Command Line

Command	Function	Input parameter
`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 9-3 Additional Commands Executable From the Command Line *(Continued)*

Command	Function	Input 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 adminmode`	Set admin mode for a KJS.	

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. The following section contains information about the following types of resource managers:

- Oracle
- Sybase

- DB2 Unix
- Microsoft SQL Server

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$)
/
```

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 9-4 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.	

Table 9-4 Error Codes *(Continued)*

Code	Cause of error	Action
04 - 08	Multiple xa_opens generate different RM ids for the same database name.	
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.	

Table 9-4 Error Codes *(Continued)*

Code	Cause of error	Action
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.
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.

Table 9-4 Error Codes *(Continued)*

Code	Cause of error	Action
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 iPlanet Application Server Administration Tool toolbar, click the Transactions button 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 the Resolve / Abort button 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 iPlanet Application Server Administration Tool 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 in a Multiple-Server Enterprise
- Distributed Data Synchronization and Load Balancing
- Multicast Communication

The Web Connector 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 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 sends the application request information for logging.

Configuring the Web Connector 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. Open the iPlanet Registry Editor by typing `kregedit` at the command line.

   The editor opens and displays the keys and values that apply to the iPlanet Application Server machine.

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. Open the iPlanet Registry Editor by typing `kregedit` at the command line.

   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 the editor tool.

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 sychronization 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” on page 195.

Configuring a Distributed Data Synchronization Environment

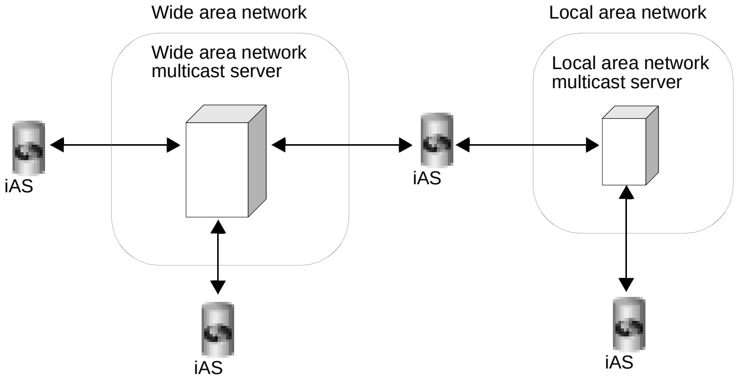
Once you install iPlanet Application Server on multiple machines, you must update the cluster keys of the servers participating in distributed data synchronization. This is done using the iPlanet Registry Editor.

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 Chapter 13, “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.

The following topics are included in this chapter:

- 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

iPlanet Application Server Administration Tool 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:

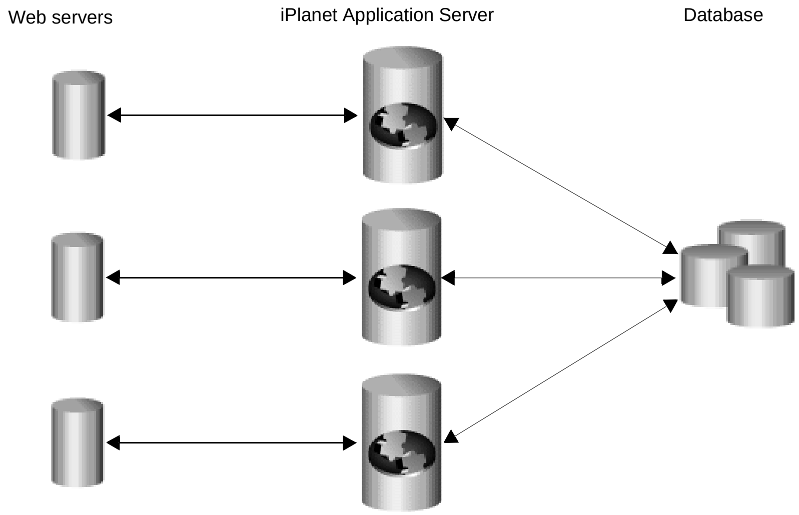
Table 11-1 Ways to Host an Application on Multiple iPlanet Application Server Machines

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.

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 Chapter 12, "Balancing User-Request Loads."

3. Disable specific application components on a server-by-server basis.

   See "Disabling and Enabling Application Components" on page 169.

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 iPlanet Application Server Administration Tool.

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) button.

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.

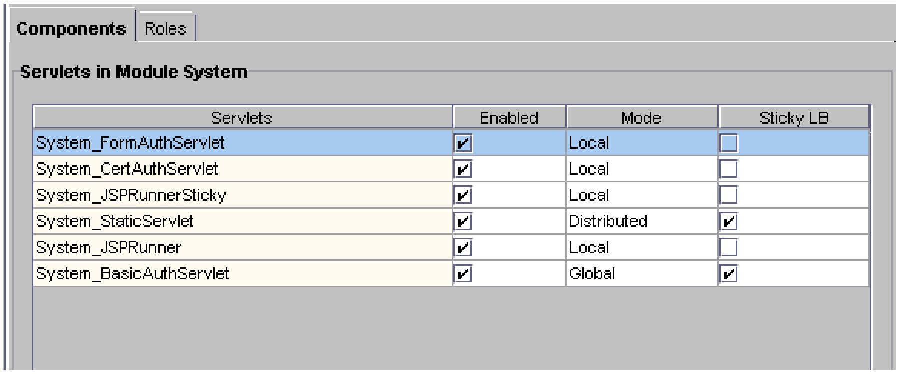
Disabling and Enabling Application Components

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 iPlanet Application Server Administration toolbar, click the Application button 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.

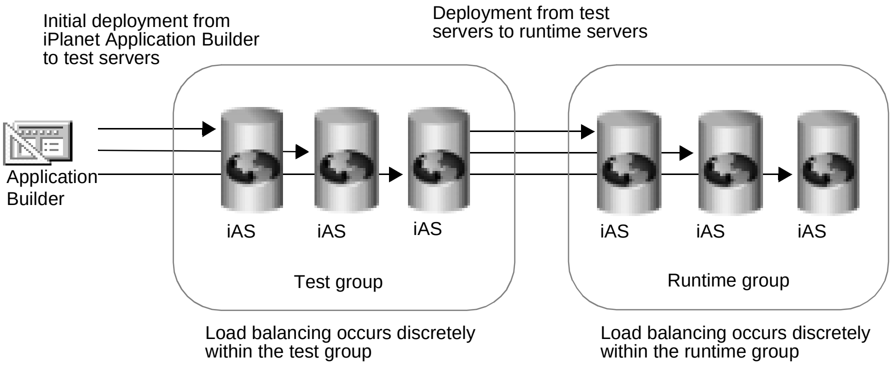
  Example 1: You might have three iPlanet Application Server machines used for testing applications. Your production environment, where users' requests are actually processed, also consists of three iPlanet Application Server machines. Because the application components could be different between the two 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.

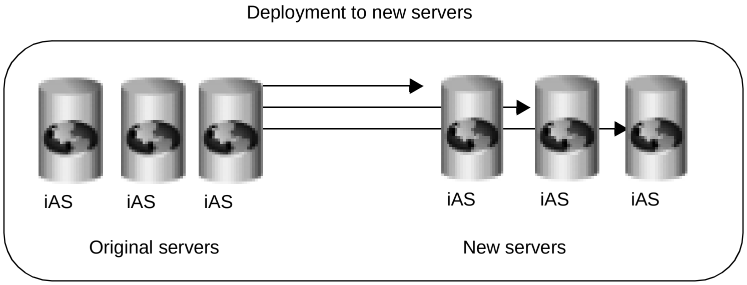
  Example 2: Suppose you scale the enterprise to include three more iPlanet Application Server machines in the production group, you can join all the servers in that group when deploying the applications from one of the existing production servers to the new servers. The application loads are then balanced 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
Application Builder
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.

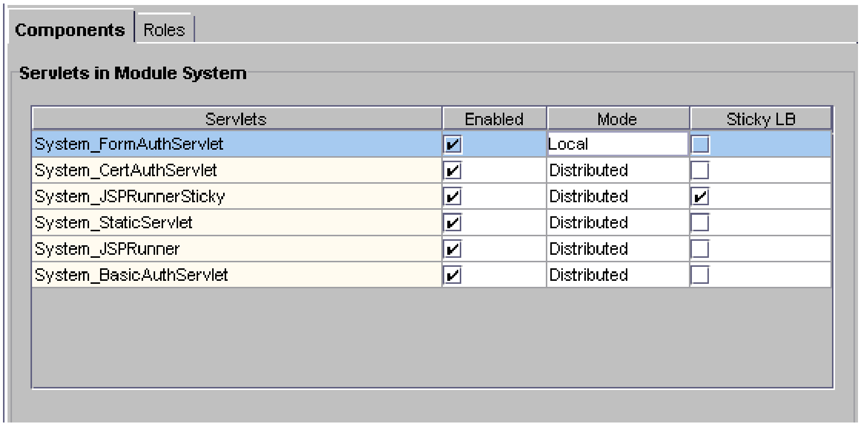
To change the distribution level for an application component, perform the following steps:

1. Open the Application window of iPlanet Application Server Administration Tool.
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 the Application Component Properties button.

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 iPlanet Application Server Administration Tool 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 driven or iPlanet Application Server driven.

- Web Connector 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 Chapter 10, "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" on page 170.

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.

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

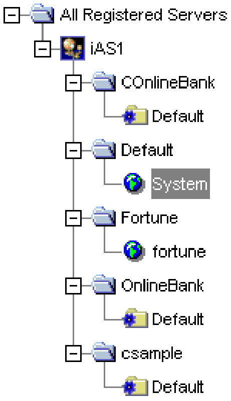
For example, a heavily used, pre-existing application is ported to run on iPlanet Application Server. Because the application is heavily used, it is distributed across several iPlanet Application Server machines to increase availability. When a user makes a request that invokes a sticky application component, 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

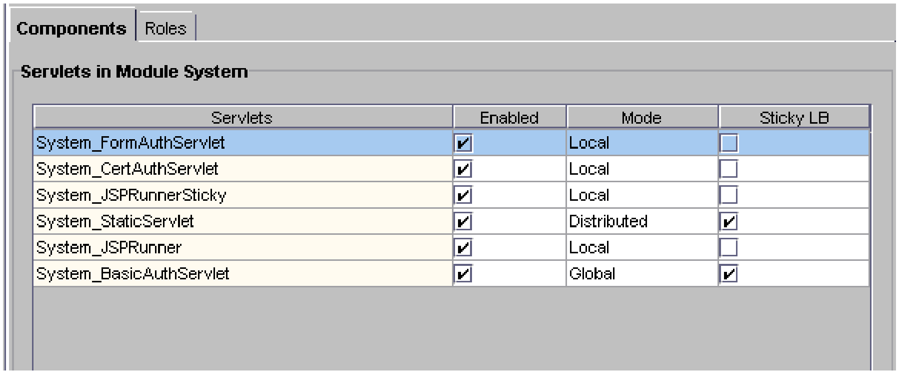
Enable sticky load balancing if there are multiple iPlanet Application Server machines and certain application components cannot distribute session and state information.

To enable sticky load balancing, perform the following steps:

1. On the iPlanet Application Server Administration Tool toolbar, click the Application button 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?

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

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 driven methods.

Per Component Response Time Load Balancing

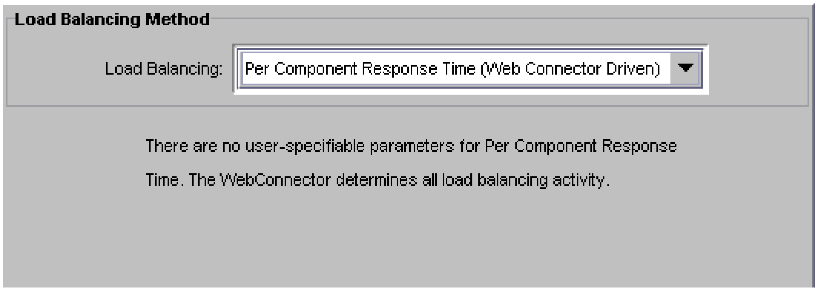
Per-component response time is based on a measure of an iPlanet Application Server machine's average response time for a specific application component.

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 iPlanet Application Server Administration Tool toolbar, click the Load Balancing button 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 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: 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

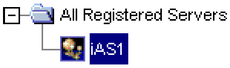
Per-server response time is based on a measure of an iPlanet Application Server machine's average response time across all the application components that machine processes.

The per-server method is best in situations where an application component has a similar response time from server to server.

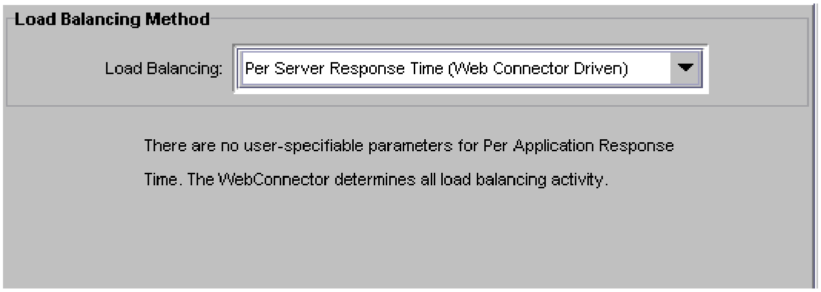
To enable per server response time load balancing, perform the following steps:

1. On the iPlanet Application Server Administration Tool toolbar, click the Load Balancing button 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 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.

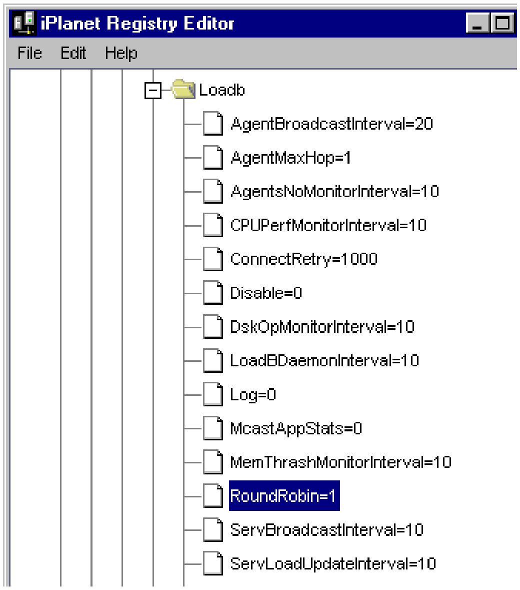
To setup round robin load balancing, perform the following steps.

1. 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.

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

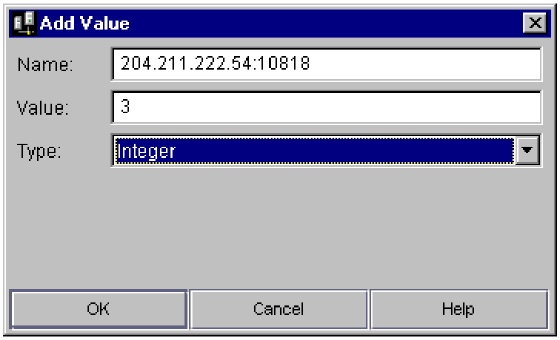
3. Highlight the following key:

   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. a. 204.211.222.54:10818

b. b. 204.211.222.56:10819

c. c. 204.211.222.59:10820

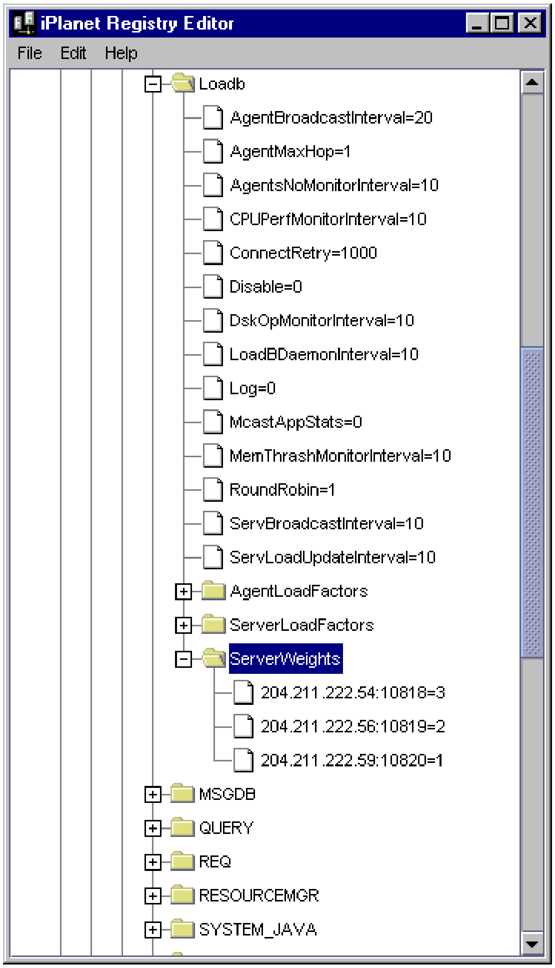
assign the following values:

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 iPlanet Application Server Administration Tool'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 iPlanet Application Server Administration Tool 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.

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.
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.

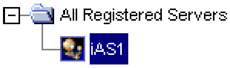
Table 12-1 Server Load Balancing Criteria

Server load criteria	Description
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.

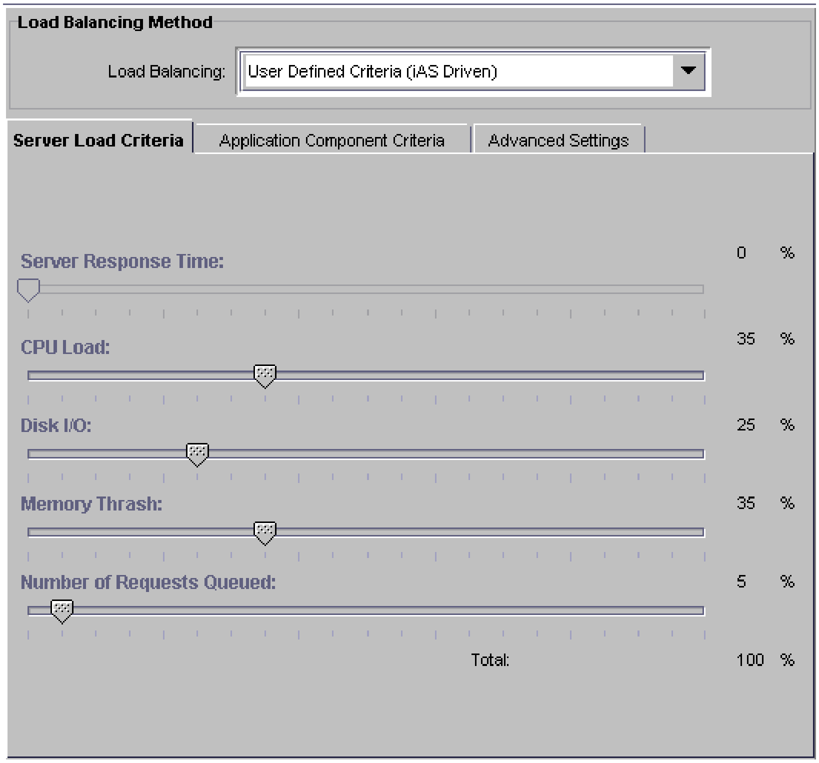
To adjust the weight factors for Server Load criteria, perform the following steps:

1. On the iPlanet Application Server Administration Tool toolbar, click the Load Balancing button 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” on page 186.

   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:

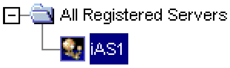
Table 12-2 Application Component Load Balancing Criteria

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 iPlanet Application Server Administration Tool toolbar, click the Load Balancing button to open the Load Balancing window.
2. In the left pane, select the server for which you want to adjust the weight factors.



All Registered Servers

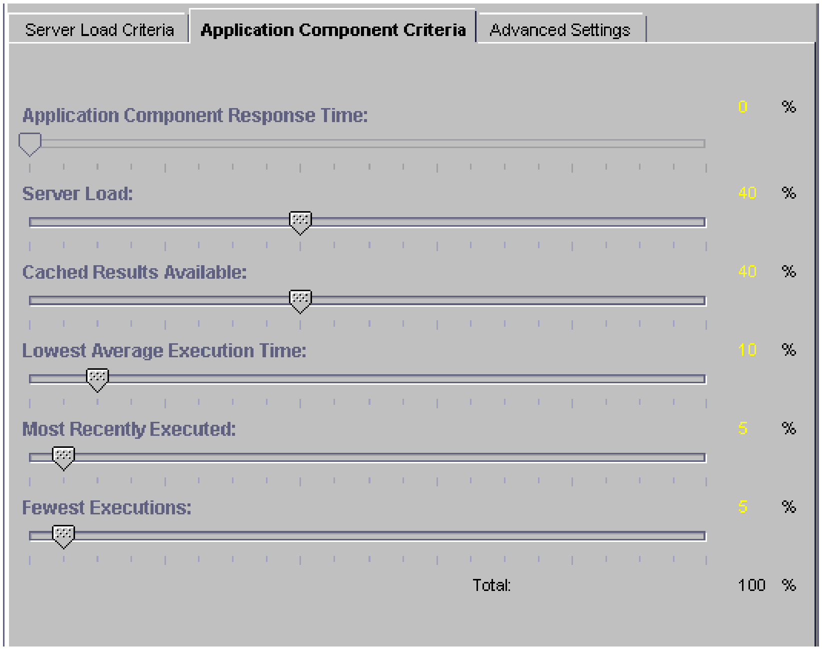
iAS1

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

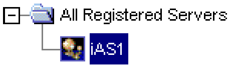
Set interval for	Description
Base Broadcast/Update Interval	The interval at which the load-balancing service “wakes up.”
Application Component Criteria	The interval at which the load-balancing service broadcasts the Application Component Performance value.
Server Load Criteria	The interval at which the load-balancing service broadcasts the Server Load value.

Table 12-3 Broadcast Intervals *(Continued)*

Set interval for	Description
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.

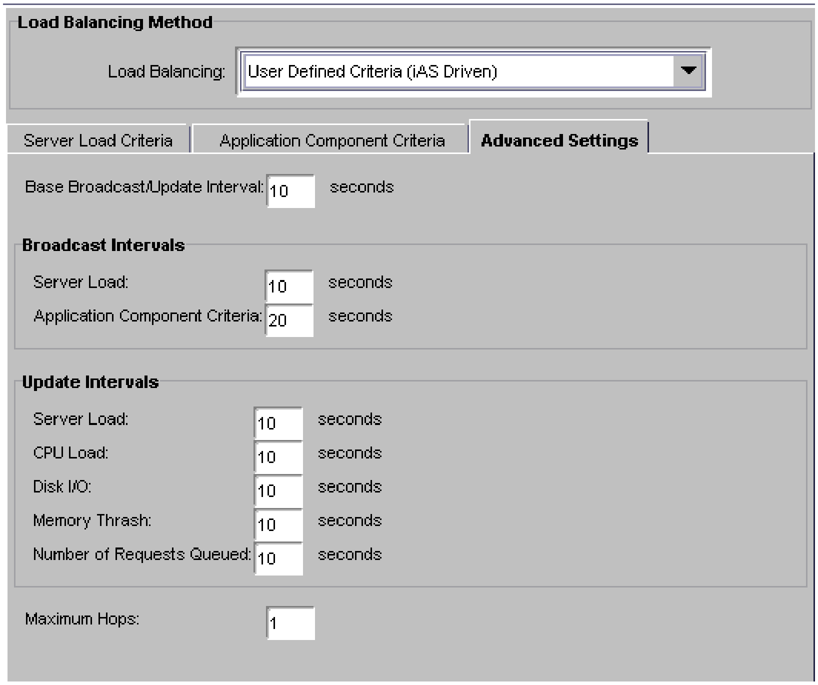
To adjust the update and broadcast intervals, perform the following steps:

1. Click the Load Balancing button on the iPlanet Application Server Administration Tool toolbar to open the 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. 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.

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 Servers into data synchronization clusters.

The following subjects are described in this chapter:

- About Distributed Data Synchronization
- How Failover Keeps Data Accessible
- What Is a Cluster?
- Setting Up and Managing Clusters
- Using the Administration Tool to Configure Clusters

About Distributed Data Synchronization

Distributed data synchronization maintains the integrity of shared state and session information across multiple iPlanet Application Server machines. This is crucial for partitioned and distributed applications that are hosted on multiple iPlanet Application Server machines.

In most enterprises, several iPlanet Application Server machines support one or more distributed applications. For such distributed applications to run successfully, each server must have access to the pertinent 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 machines.

Each iPlanet Application Server machine is made up of the following four “engines:”

- Administrative Server (KAS) – An 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 machines, 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 the Java or C++ engine on an iPlanet Application Server fails, the Administrative Server simply restarts the KJS or KCS. However, if the Executive Server fails, the Administrative Server performs the following actions:

  - Brings the Executive Server back up in the currently appropriate role. This role is determined in synchronization with other Executive Servers in the cluster, and is not necessarily the previous role.
  - Brings down the Java and C++ engines.
  - Brings the Java and C++ engines back up.

What Is a Cluster?

A cluster is a group of iPlanet Application Server machines that synchronizes data. Servers in a cluster are connected by the same network.

Data that is shared by all the iPlanet Application Server machines in a cluster is stored in iPlanet Directory Server. Each iPlanet Application Server machine in your cluster should share one Directory Server; if the iPlanet Application Server machines 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 defeats the purpose of Directory Server, which is designed to simplify information storage by storing the data shared by servers in your enterprise in a central location.

NOTE You access cluster information using the iPlanet Registry Editor. You cannot edit an iPlanet Application Server machine's cluster settings using the Windows NT regedit tool or any other editor tool. Each folder in the iPlanet Registry Editor tree structure, which looks similar to Windows NT's registry tree structure, is referred to as a kregedit key or cluster key in this document.

Setting Up Data Synchronization

To set up data synchronization between servers, you must first decide what general role each server performs in the cluster. Then you can edit each cluster entry to set up the server roles and to register the cluster with the synchronizer service. Finally, start each iPlanet Application Server in the order that is determined by server roles.

Synchronization Server Roles

Each server that participates in data synchronization can be set up to fill any one of the roles described in the following table.

Table 13-1 Roles for Data Synchronization

Server role	Description
Sync Server	Any iPlanet Application Server machine that can potentially become a 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 kregedit.

Table 13-1 Roles for Data Synchronization *(Continued)*

Server role	Description
Sync Primary	The server that is the primary data store, to 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.
Sync Alternate	A server listed in the `SyncServers` kregedit key 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.

Table 13-1 Roles for Data Synchronization *(Continued)*

Server role	Description
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` kregedit key. 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` kregedit key 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` kregedit key 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

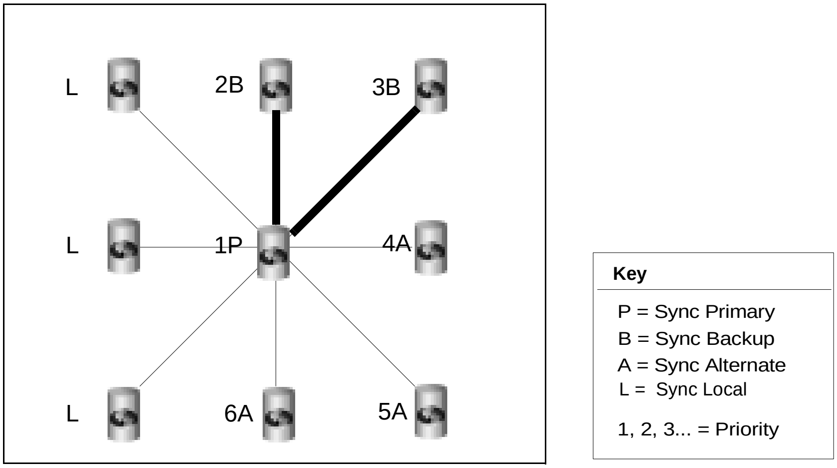
Servers in a cluster communicate using the GXCONN communication protocol. However, before the servers in a cluster can communicate with each other, each server has to know what cluster it belongs to. iPlanet Application Server becomes an active part of a cluster when you map its synchronizer to the cluster. This procedure is described in “Mapping the Synchronizer to the Cluster” on page 210.

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.

Although you can define as many clusters as you like, the synchronizer for each iPlanet Application Server machine can be mapped 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, although the Sync Backups communicate with it most closely. Also, notice that no Sync Local is assigned a priority number.

Note also that the illustration is an ideal representation of a cluster that has probably just started and has not experienced failover, in that the priority numbers correspond gracefully with the currently assigned roles.

Setting Up and Managing Clusters

Before you set up and begin managing clusters, review the following steps, which provide an overview of the general procedure. More specific procedures for setting up and managing clusters are described in subsequent sections.

1. Decide which servers will participate in a synchronization group (cluster), and which of those servers will be Sync Servers, eligible to act as the Sync Primary and as Sync Backups, and which will be Sync Locals.

2. Edit the kregedit keys under `Clusters` and `ClusterName` on one of the Sync Servers. Duplicate the `ClusterName` edits to the registries of all the other servers in the cluster (including the Sync Locals). You need not duplicate edits to the `Clusters` key since this information is stored in a centrally located Directory Server.

    a. Create the kregedit keys that will contain synchronization information, if necessary.

    b. Edit the `SyncServers` kregedit key to contain identification information and the priority setting for each Sync Server in the cluster. Often, the larger and more powerful servers are chosen to be the highest-priority Sync Servers.

    c. Set the `MaxBackups` kregedit key to the number of Sync Backups. Sync Backups are servers that duplicate the data on the Sync Primary.

3. Enter the name of the cluster in the `ClusterName` key.

    Make sure that the kregedit keys under `ClusterName` are identical on all servers in the cluster, including the Sync Locals. Each `SyncServers` kregedit key must list the same Sync Servers with the same priority numbers, or the cluster will not function properly.

4. Start the Sync Server that will be the Sync Primary. The server that you want to be the Sync Primary must always be the first server to be started in the cluster, and it becomes the Sync Primary simply because it started first.

5. After starting the Sync Primary, start the other servers (including the Sync Locals). Although the starting order is not mandated after the Sync Primary starts, it is a good practice to start the Sync Servers in priority order, and then to start the Sync Locals.

    a. Start the servers that will become the Sync Backups, up to the value of `MaxBackups`. By default, the next servers listed in the `SyncServer` key that start, up to the value stored under the `MaxBackups` kregedit key, will become the Sync Backups.

    b. After `MaxBackups` number of servers have started, remaining Sync Servers that start become Sync Alternates.

    c. All servers not listed in the `SyncServers` kregedit key become Sync Locals. Sync Locals are part of the cluster simply because each is mapped to the cluster and the `SyncServers` kregedit key on each contains a list of all the Sync Servers in the cluster.

Determining Sync Server Priority

The specific procedure for setting priority is covered in “Modifying the Default Cluster for Fast Cluster Setup” on page 205 and “Defining a Cluster” on page 211. The following section discusses general priority issues and gives a comprehensive example of cluster coordination.

Priority is indicated by an integer value that is set in the `SyncServers` kregedit key. The lower the value, the higher the priority, so the server assigned a value of 0 has the highest possible priority. The highest acceptable value, and so the lowest priority value, is 65,535.

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 before any of the other servers. 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 machine 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 machines 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 two.

- 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 two, 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.

Modifying the Default Cluster for Fast Cluster Setup

The fastest and easiest way to set up a cluster is during installation.

After installation, the easiest way to set up a cluster is to modify the default cluster that was automatically created when you installed iPlanet Application Server. At installation, the `SyncServers` kregedit key for the default cluster lists only one server—the server itself. The default cluster is the name of *`hostname`*`-NoDsync`, where *`hostname`* is the name of your local machine. For instance, if you install iPlanet Application Server on a machine named “pc543714,” the default cluster is `pc543714-NoDysnc`. The default cluster contains all that a cluster needs to be complete and active except for the new name for the cluster and the names of all Sync Servers with which to synchronize.

Because the default cluster already contains all the kregedit keys that a cluster needs, you can easily set up a cluster by making a few substitutions in the kregedit keys for the default cluster. If you were creating a completely new cluster, you would have to create the kregedit keys for that new cluster.

Entering IP Addresses Using kregedit

When you edit the `SyncServers` key for the default cluster, you will enter the IP address for each of the Sync Servers in your cluster.

At installation, the IP address for each iPlanet Application Server machine is placed in the `SyncServers` key of that server’s default cluster. When you enter the address for each Sync Server into the first `SyncServer` registry key, remember that you can find the information in the registry for each iPlanet Application Server machine.

Note, however, that you will remove this entry on each Sync Local. If you decide later to promote a Sync Local to a Sync Server, you will have to find the address information elsewhere.

Editing Default Cluster Keys

Sync Locals are never listed in the `SyncServers` key for a legitimate cluster. But, because each Sync Local is automatically listed in its own default cluster, you must remove each Sync Local from its own `SyncServers` key.

This necessity will be obvious if the cluster settings you edit belong to a Sync Server.

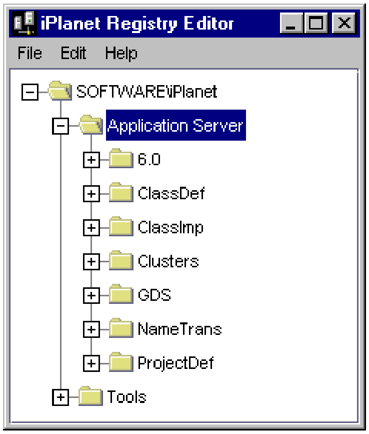
To edit the default cluster keys, perform the following steps:

1. Stop the application server whose settings you will edit.

Be aware 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.

2. Open kregedit by typing `kregedit` at the command prompt.

   The kregedit tool displays the keys and values that apply to the iPlanet Application Server machine as shown in the following illustration:

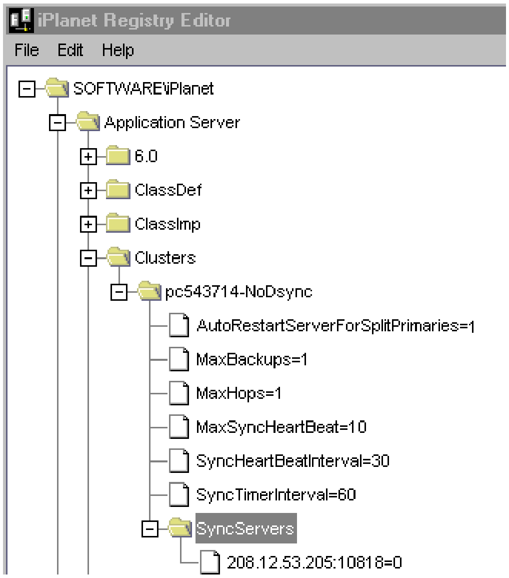


3. Open the following folder:

```
SOFTWARE\iPlanet\Application Server\Clusters\
```

   In this example, the default cluster is named `pc543714-NoDsync` and, so far, contains one Sync Server with a priority of zero.

```
SOFTWARE\iPlanet\Application
 Server\Clusters\pc543714-NoDsync\SyncServers
```



```
SOFTWARE\iPlanet
    Application Server
        6.0
        ClassDef
        ClassImp
        Clusters
            pc543714-NoDsync
                AutoRestartServerForSplitPrimaries=1
                MaxBackups=1
                MaxHops=1
                MaxSyncHeartBeat=10
                SyncHeartBeatInterval=30
                SyncTimerInterval=60
                SyncServers
                    208.12.53.205:10818=0
```

Whenever one of the following steps directs you to modify a key name or value, you can modify that name or value by performing the following steps:

a. Double-click the key name to bring up the Modify Value dialog box.

b. Enter the new name or value in the dialog box.

c. Click OK.

4. Change the default name (in this case, `pc543714-NoDsync`) to a new, unique name for your cluster.

5. Modify the `AutoRestartServerForSplitPrimaries`, as necessary.

`SOFTWARE\iPlanet\Application Server\Clusters\`*`hostname`*`-NoDsync`

If set to true (1) then, when the heartbeat mechanism detects a split-primary (dual-primary) server role, the server with the lower priority among the two Sync Primaries is automatically restarted and the abnormal cluster condition is automatically corrected. See step 8 for more information about the heart beat mechanism.

NOTE You can also set this on the Cluster tab of the General window of the iPlanet Application Server Administration Tool. See "Setting Cluster Parameters" on page 222 for more information.

Note, you can also set this on the Cluster tab of the General window of the iPlanet Application Server Administration Tool.

6. Check and modify the `MaxBackups` value, as necessary.

   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?" on page 196.

   Because all Sync Backups are updated at the same time, an extra load is created for each additional backup server. Consider the performance impact when you set the number of backups, and try to 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 Ignore the `MaxHops` key. This key relates to an unsupported feature and is now ignored by the server.

7. Check and modify the `MaxSyncHeartBeat` value, as necessary.

   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. Abnormal cluster conditions are defined double-primary (split-primary) and no primary server roles.

   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. See step 6 for more information.

8. Check and modify the `SyncHeartBeatInteveral` value, as necessary.

   This value specifies the number of seconds between two heartbeat messages sent from one server to another. The default value is 30 seconds.

9. Check and modify the `SyncTimerInterval` value, as necessary, which is found in the following location:

   `SOFTWARE\iPlanet\Application Server\Clusters\`*`hostname`*`-NoDsync`

   This key specifies the intervals, in seconds, at which the synchronization service wakes up and checks to see whether any data has expired. Specifically, 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.

10. Add each Sync Server to the cluster under `SyncServers`.

    The IP addresses and port numbers under the `SyncServers` key are the Executive Server processes of the iPlanet Application Server machines that belong to this cluster. Each server is listed by its `host IP address:KXS port number=priority level`.

    a. Add the IP address and port number for the Sync Server.

    b. Set the priority for each Sync Server by double-clicking the priority value to bring up a pop-up window, entering the priority number, and clicking OK. The IP address, port number, and priority for the Sync Server should have been listed under the `SyncServers` key at installation.

    The priority setting for a data synchronization server determines which Sync Backup in a group of Sync Backups will become the replacement Sync Primary, and which Sync Alternate in a group of Sync Alternates will become the replacement Sync Backup.

    Priority settings start at zero, the highest priority setting. The lowest priority is 65,535. For more information about priority, see “Determining Sync Server Priority” on page 202.

11. Close kregedit when you have finished.

12. Restart all application servers effected by these modifications.

    All changes you make to the `SyncServers` key now apply to each server in the cluster

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

For a cluster to communicate, the synchronizer in each server must know to which cluster the synchronizer belongs. This is done by mapping the `ClusterName` key of each synchronizer to the name of an actual cluster.

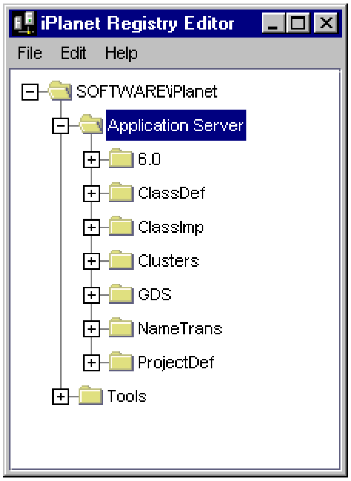
To map the synchronizer to a cluster, perform the following steps:

1. Stop the application server whose registry you will edit.

   Be aware that editing the server registry while iPlanet Application Server is running can cause serious problems. Also, some changes take effect only after the engine is recycled.

2. Open kregedit by typing `kregedit` at the command prompt.

   The kregedit tool displays the keys and values that apply to the iPlanet Application Server machine.

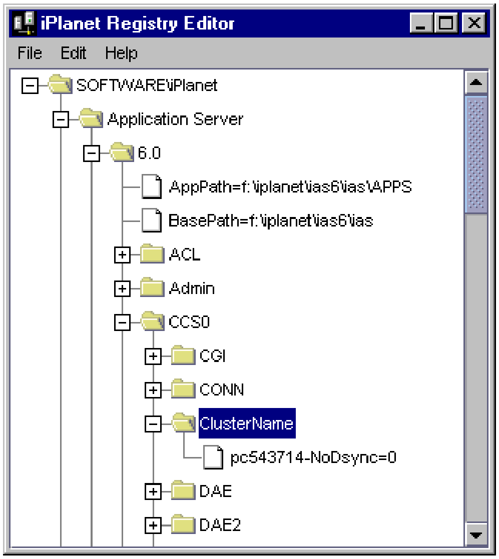
   

3. Open the following key:

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

   The following example shows the default cluster that has already been renamed "SampleCluster."

```
SOFTWARE\iPlanet\Application
 Server\6.0\CCS0\ClusterName\hostname-NoDsync=0
```



```
SOFTWARE\iPlanet
  Application Server
    6.0
      AppPath=f:\iplanet\ias6\ias\APPS
      BasePath=f:\iplanet\ias6\ias
      ACL
      Admin
      CCS0
        CGI
        CONN
        ClusterName
          pc543714-NoDsync=0
        DAE
        DAE2
```

4. Rename the key under `ClusterName` to the name of the cluster to which the synchronizer should connect.

   If this key has not been previously modified, then the name under `ClusterName` will be *`hostname`*`-NoDsync`, where *`hostname`* is the name of your local machine.

5. Close kregedit when you are finished. The synchronizer should now be mapped to the cluster

Defining a Cluster

Create a cluster to organize iPlanet Application Server machines into data-synchronizing network-centric groups.

Even though each iPlanet Application Server machine can be mapped to only one cluster at a time, you can define as many clusters as you like. Some installations might define multiple clusters for testing purposes, for example.

While you can edit the default cluster to easily set up your first cluster definition, editing the default cluster defines only one cluster. To get more than one definition, create the additional clusters.

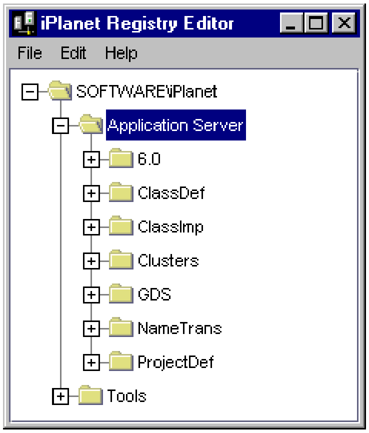
To create a cluster, perform the following steps:

1. Stop the application server whose settings you will edit.

   Be aware 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.

2. Open kregedit by typing `kregedit` at the command prompt.

   The kregedit tool displays the keys and values that apply to the iPlanet Application Server machine as shown in the following illustration:

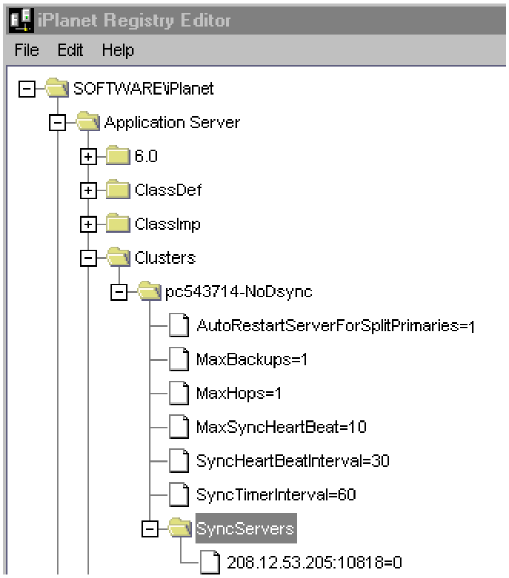


3. Open the following folder:

   ```
   SOFTWARE\iPlanet\Application Server\Clusters\
   ```

   In this example, the default cluster is named `pc543714-NoDsync` and, so far, contains one Sync Server with a priority of zero.

   ```
   SOFTWARE\iPlanet\Application
    Server\Clusters\pc543714-NoDsync\SyncServers
   ```



```
SOFTWARE\iPlanet
    Application Server
        6.0
        ClassDef
        ClassImp
        Clusters
            pc543714-NoDsync
                AutoRestartServerForSplitPrimaries=1
                MaxBackups=1
                MaxHops=1
                MaxSyncHeartBeat=10
                SyncHeartBeatInterval=30
                SyncTimerInterval=60
                SyncServers
                    208.12.53.205:10818=0
```

Whenever one of the following steps directs you to modify a key name or value, you can modify that name or value by performing the following steps:

a. Double-click the key name to bring up the Modify Value dialog box.

b. Enter the new name or value in the dialog box.

c. Click OK.

4. Change the default name (in this case, `pc543714-NoDsync`) to a new, unique name for your cluster.

5. Check and modify the `AutoRestartServerForSplitPrimaries,` as necessary.

`SOFTWARE\iPlanet\Application Server\Clusters\`*`hostname`*`-NoDsync`

If set to true (1) then, when the heartbeat mechanism detects a split-primary (dual-primary) server role, the server with the lower priority among the two Sync Primaries is automatically restarted and the abnormal cluster condition is automatically corrected. See step 8 for more information about the heart beat mechanism.

6. Check and modify the `MaxBackups` value, as necessary.

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?” on page 196.

Because all Sync Backups are updated at the same time, an extra load is created for each additional backup server. Consider the performance impact when you set the number of backups, and try to 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 Ignore the `MaxHops` key. This key relates to an unsupported feature and is now ignored by the server.

---

7. Check and modify the `MaxSyncHeartBeat` value, as necessary.

   This value specifies the maximum number of heartbeat messages that an engine will send to any other engine. The heartbeat mechanism is used to detect an abnormal cluster condition. Abnormal cluster conditions are defined double-primary (split-primary) and no primary server roles.

   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. See step 6 for more information.

8. Check and modify the `SyncHeartBeatInteveral` value, as necessary.

   This value specifies the number of seconds between two heartbeat messages sent from one server to another. The default value is 30 seconds.

9. Check and modify the `SyncTimerInterval` value, as necessary, which is found in the following location:

SOFTWARE\iPlanet\Application Server\Clusters\hostname-NoDsync

This key specifies the intervals, in seconds, at which the synchronization service wakes up and checks to see whether any data has expired. Specifically, 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.

10. Add each Sync Server to the cluster under `SyncServers`.

    The IP addresses and port numbers under the `SyncServers` key are the Executive Server processes of the iPlanet Application Server machines that belong to this cluster. Each server is listed by its `host IP address:KXS port number=priority level`.

    a. Add the IP address and port number for the Sync Server.

    b. Set the priority for each Sync Server by double-clicking the priority value to bring up a pop-up window, entering the priority number, and clicking OK. The IP address, port number, and priority for the Sync Server should have been listed under the `SyncServers` key at installation.

    The priority setting for a data synchronization server determines which Sync Backup in a group of Sync Backups will become the replacement Sync Primary, and which Sync Alternate in a group of Sync Alternates will become the replacement Sync Backup.

    Priority settings start at zero, the highest priority setting. The lowest priority is 65,535. For more information about priority, see “Determining Sync Server Priority” on page 202.

11. Close kregedit when you have finished.

12. Restart all application servers effected by these modifications.

    All changes you make to the `SyncServers` key now apply to each server in the cluster

After correctly completing these steps, you have defined a cluster. You can define as many clusters as you like, but you can map the synchronizer to only one cluster at a time. See “Mapping the Synchronizer to the Cluster” on page 210 for the procedure that enables communication.

Using the Administration Tool to Configure Clusters

You can perform the following tasks to configure clusters using iPlanet Application Server Administration Tool:

- Creating a Cluster
- Adding a Server to a Cluster
- Removing a Server from a Cluster
- Changing Sync Server Priority
- Modify the maximum number of Sync Backups
- Setting Cluster Parameters

Note that to properly configure a cluster using iPlanet Application Server Administration Tool, you must register all the servers in the cluster. Otherwise, configuration changes will not apply across all the servers in the cluster.

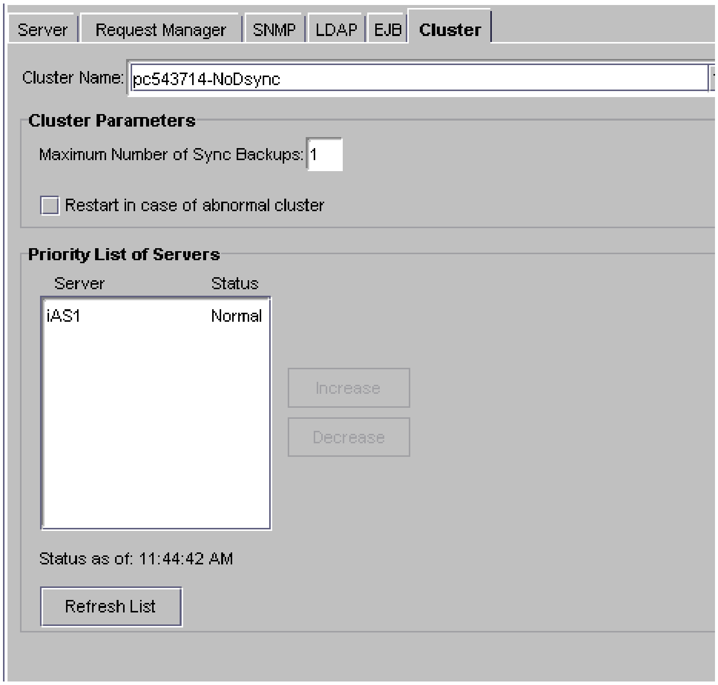
For information about editing cluster settings directly, see the various sections earlier in this chapter that discuss how to configure clusters.

Creating a Cluster

To create a new cluster, perform the following steps:

1. From the iPlanet Application Server Administration Tool toolbar, click the General button to open the General window.
2. In the right pane of the General window, click the Cluster tab.

   The following window appears:



Cluster Name: pc543714-NoDsync

Maximum Number of Sync Backups: 1

□ Restart in case of abnormal cluster

Server	Status
iAS1	Normal

Status as of: 11:44:42 AM

3. Highlight the Cluster Name to select *`hostname`*`-No-Dsync`.

4. Use the Delete key on your keyboard to clear the Cluster Name drop-down box.

5. Type the name of your new cluster in the Cluster Name drop-down box and press the Enter key on your keyboard.

   You can choose any unique name for the new cluster.

6. Click Apply Changes.

   Your changes do not take effect until you restart the server.

After you restart the server, you can add iPlanet Application Server machines to the new cluster as described in the following section.

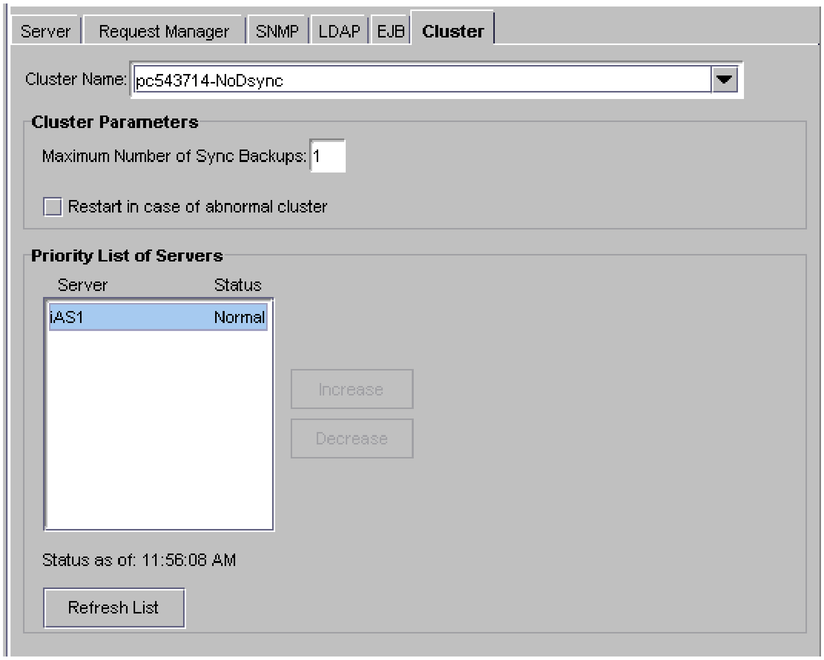
Adding a Server to a Cluster

To add an unassigned server to a cluster, or to reassign a server to a different cluster, perform the following steps:

1. From the iPlanet Application Server Administration Tool toolbar, click the General button to open the General window.

2. Click the Cluster tab.

   The following window appears:



Cluster Name: pc543714-NoDsync

Cluster Parameters

Maximum Number of Sync Backups: 1

□ Restart in case of abnormal cluster

Priority List of Servers

Server	Status
iAS1	Normal

Status as of: 11:56:08 AM

   A list of all 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 as shown in the previous illustration.

   The Priority List of Servers box also shows the cluster status of the server. Server conditions can be Normal, Dual Primary or No Primary. You can 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.

   A server that is not a member of a cluster, hence not participating in data synchronization, is listed under *`hostname`*`-NoDsync`, in the cluster list on the right.

4. From the Cluster Name drop-down box, select the name of the cluster you want to add the server to.

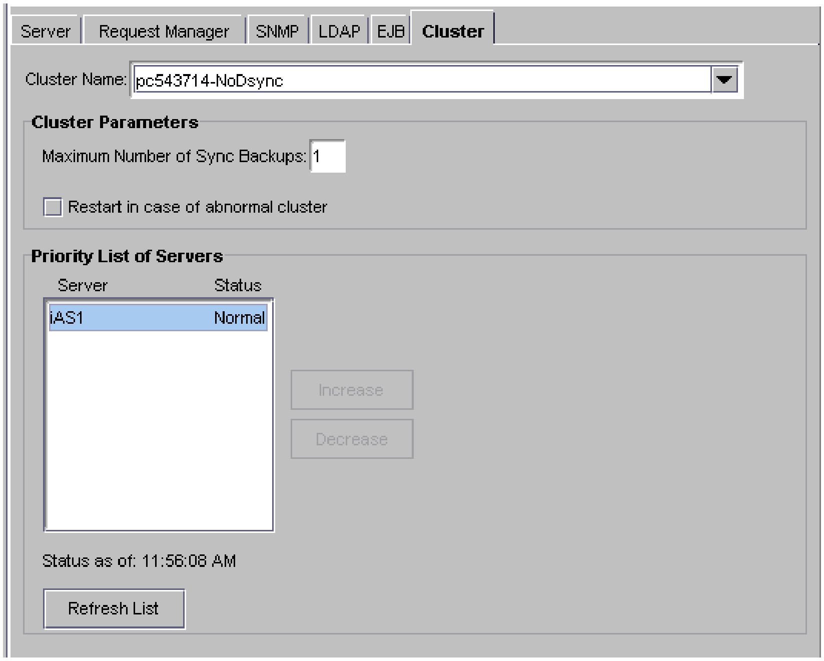
The Cluster Name drop-down list is populated with the cluster names that all *registered* servers belong to. If the servers in a cluster are not registered by iPlanet Application Server Administration Tool, then that cluster does not appear in the Cluster Name drop-down box. For the name of a cluster to appear in Cluster Name, you must register one or more of the servers in that cluster.

5. Click Apply Changes.
6. Shut down and restart every server in the cluster, including the server you just added. For changes to apply across the cluster, you must restart every machine to reload the memory on each machine with the cluster configuration changes. If at least one machine has a different cluster configuration loaded into memory than the other machines in the cluster, the new settings will not take effect and data synchronization will not work properly.
7. If when adding the server to a cluster, you removed it from another, you must also shut down and restart every server in the cluster from which it was removed.

Removing a Server from a Cluster

To remove a server from a cluster, perform the following steps:

1. From the iPlanet Application Server Administration Tool toolbar, click the General button to open the General window.
2. Click the Clusters tab to display the following window:



Cluster Name: pc543714-NoDsync

Cluster Parameters

Maximum Number of Sync Backups: 1

□ Restart in case of abnormal cluster

Priority List of Servers

Server	Status
iAS1	Normal

Status as of: 11:56:08 AM

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.

3. In the left pane of the General window, click the name of the server you want to remove from the cluster.

   You can remove a server from a cluster only when it is assigned to a cluster and registered with iPlanet Application Server Administration Tool. A server that is not a member of a cluster, hence not participating in data synchronization, is listed under *`hostname`*`-NoDsync`, in the cluster list. You cannot remove a server from the *`hostname`*`-NoDsync` list.

   Note that you cannot remove an unregistered server from a cluster.

4. Click Remove from Cluster.

5. Click Apply Changes.

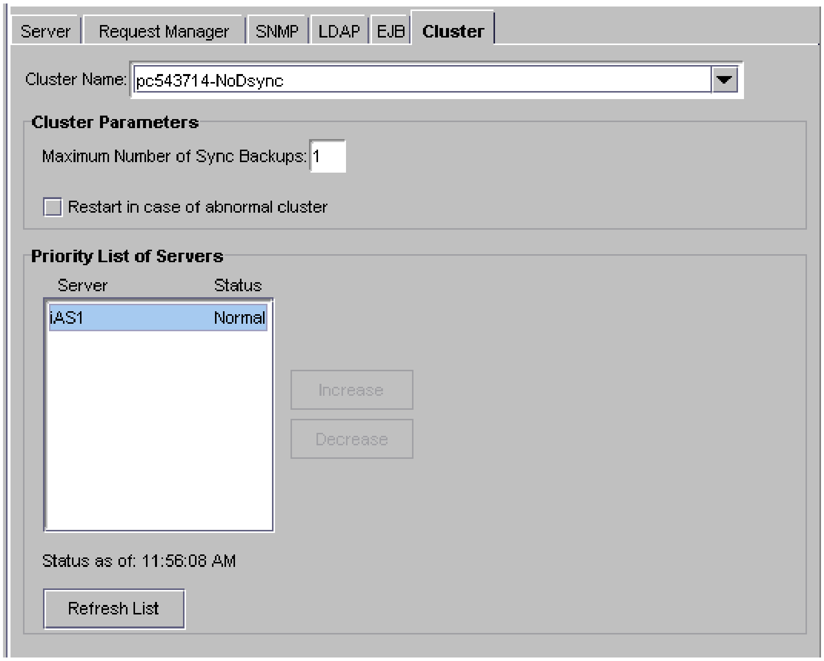
6. Shut down and restart every server in the cluster, including the server you just removed. For changes to apply across the cluster, you must restart every machine to reload the memory on each machine with the cluster configuration changes. If at least one machine has a different cluster configuration loaded into memory than the other machines in the cluster, the new settings will not take effect and data synchronization will not work properly.

Changing Sync Server Priority

To assign a new Sync Server priority to a server that is in a cluster, perform the following steps:

1. From the iPlanet Application Server Administration Tool toolbar, click the General button to open the General window.

2. Click the Clusters tab to display the following window:



Cluster Name: pc543714-NoDsync

Cluster Parameters

Maximum Number of Sync Backups: 1

□ Restart in case of abnormal cluster

Priority List of Servers

Server	Status
iAS1	Normal

Status as of: 11:56:08 AM

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 machine in the cluster. The status should always be "Normal." If an abnormal cluster condition exists, it could also show "Dual Primary" or "No Primary." To ensure that these conditions are corrected, see "Setting Cluster Parameters" on page 222. The Refresh List button causes iPlanet Application Server Administration Tool to immediately check for status. Normally it checks every 15 seconds.

3. 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.

4. In the Priority List text box, click the name of the server whose Sync Server priority you want to change.

   You can change Sync Server priority order only for a registered server that belongs to a cluster. A server that is not a member of a cluster, hence not participating in data synchronization, is listed under *`hostname-NoDsync`*, in the cluster list on the right.

5. Click one of the following:

   - Increase to assign a higher priority.
   - Decrease to assign a lower priority.

   Click as many times as you want to increase or decrease the priority. 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.

6. When you finish reassigning priorities, click Apply Changes.

7. 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 every machine so that they are all aware of their new priority sequence, relative to one another.

Setting Cluster Parameters

You can set the following cluster parameters:

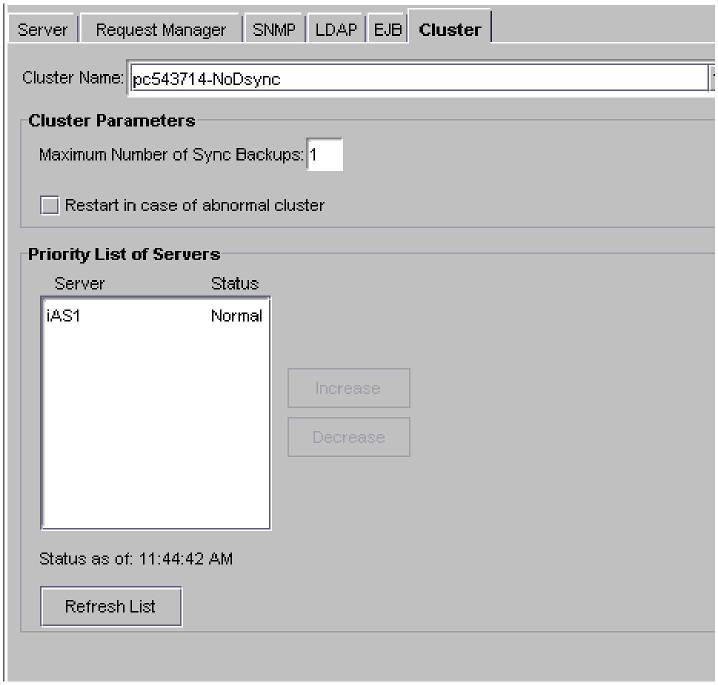
- Maximum Number of Sync Backups
- Restart in case of abnormal cluster

You can specify the maximum number of Sync Backups the Sync Primary will use. In clusters of numerous machines, this allows you to control how many other machines are used as backups.

You can also enable the appropriated process to restart in case an abnormal cluster condition is detected. An abnormal cluster condition is either a cluster that has more than one iPlanet Application Server machines with the Sync Primary (dual-primary) role or no iPlanet Application Server machines with the Sync Primary role.

To set the cluster parameters, perform the following steps:

1. From the iPlanet Application Server Administration Tool toolbar, click the General button to open the General window.
2. Click the Clusters tab to display the following window:



Cluster Name: pc543714-NoDsync

Cluster Parameters

Maximum Number of Sync Backups: 1

□ Restart in case of abnormal cluster

Priority List of Servers

Server	Status
iAS1	Normal

Status as of: 11:44:42 AM

3. In the left pane of the General window, click the name of a server that is a member of the cluster you want to modify.
4. Enter the maximum number of Sync Backups allowed during a single cluster session in Maximum Number of Sync Backups.
5. Check the Restart in case of abnormal cluster 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 every machine so that they are all aware of the changes.

Appendix A

Troubleshooting

This appendix contains the following information about troubleshooting iPlanet Application Server:

- Configuring the Class Path
- Setting up Transactions
- Setting Environment Variables for Databases

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. Refer to “Setting Up Resource Managers for Distributed Transactions” on page 142 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” on page 152.

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 `init`*`instancename`*`.ora` file in the `pfile` directory of the Oracle server installation.

Setting Environment Variables for Databases

See “Post-Installation Notes” in the *Installation Guide*.

Index