

Deployment Guide

iPlanet Market Maker

Version 4.5

March 2002

Copyright © 2002 Sun Microsystems, Inc. All rights reserved.

Sun, Sun Microsystems, the Sun logo, iPlanet, Java and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation. No part of the product or this document may be reproduced in any form by any means without prior written authorization of the Sun Microsystems, Inc. and its licensors, if any.

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.

Copyright © 2002 Sun Microsystems, Inc. Tous droits réservés.

Sun, Sun Microsystems, et Sun logo, iPlanet, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et d'autre pays.

Le produit décrit dans ce document est distribué selon des conditions de licence qui en restreignent l'utilisation, la copie, la distribution et la décompilation. Aucune partie de ce produit ni de ce document ne peut être reproduite sous quelque forme ou par quelque moyen que ce soit sans l'autorisation écrite préalable de Sun Microsystems, Inc. et, le cas échéant, de ses bailleurs de licence.

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

Contents

List of Figures	7
List of Tables	9
About This Document	11
Audience	11
What's in This Document	12
Documentation Conventions	12
The Document Online	13
Related Documentation	13
Other Useful Documentation	14
Product Support	14
Chapter 1 Planning the Deployment	17
Overview of iPlanet Market Maker	17
Overview of the Deployment Process	20
Planning for Installation	22
Software Requirements	23
Patches	24
iPlanet Patch Information	24
Solaris Patch Information	25
Developing the Deployment Plan	25
Determining the Development Setup	25
Basic Development Setup	26
Advanced Development Setup	26
Determining the Marketmaker Requirements	27
Estimating Initial Hardware Requirements	28

Determining the Production Setup	28
Basic Production Setup	28
Production Setup with SSL	29
High-Availability Production Setup	29
Hardware Configurations	30
Hardware Components	31
Basic Production Setup	32
Production Setup With SSL	33
High-Availability Production Setup	35
Summary	37
Chapter 2 Sizing Your System	41
Overview of the Sizing Calculator	41
Example 1: Auction-Intensive Marketplace	44
Example 2: RFx-Intensive Marketplace	45
Example 3: Exchange Intensive Marketplace	46
Chapter 3 Configuring Your System	47
Configuring Oracle	47
Setting Oracle Initialization Parameters	47
Oracle Initialization Parameters	48
Configuring Net8	49
Configuring iPlanet Market Maker for the Databases	49
Connecting to Remote LDAP Instances	50
To connect to a remote LDAP instance	50
LDAP Connection Pool Settings	50
Connecting to Remote Oracle Instances	51
Oracle Connection Pool Settings	51
Configuring the Schemas	51
Sizing Objects (Tables and Indexes)	54
Re-installing the Database Schema Using the Installer	54
Re-installing the Database Schema Using Scripts	54
Catalog Module	55
Auctions Module	56
Exchange Module	56
In-Box Notification	57
Setting the Job Scheduler	58
Maximum Heap Size for the JV	58
OnceOnly and Repeatable Timer Thread Pool Sizes	58
OnceOnly Worker Thread Pool Size	59
Repeatable Worker Thread Pool Size	59
Sleep Interval for OnceOnly Thread	60

Repeat Sleep Interval	60
Transactional Mail Scheduler Interval	61
Max Threads in Notifier Pool	61
Importing and Configuring Data	62
Creating the LDAP Schema	63
Deleting the Existing LDAP Data	63
Creating the Oracle Schema	64
Running iPlanet Market Maker Software Over SSL (Secure Sockets Layer)	65
Viewing HTTPS Sites (External URLs) in the iPlanet Market Maker Software	65
Install and Customize the iPlanet Market Maker Software for Secure Site Access	65
Download JSSE 1.0.2 from the Sun Site	65
Install JSSE Pack	66
Install the Certificate of the Secure Site	66
Set the Security Provider	66
Set the Configuration Parameters	67
Chapter 4 Performance Tuning and Monitoring	69
Testing Scenarios	69
Tuning the System	70
Tuning Solaris Kernel Parameters	70
Configuring Memory	71
Configuring Disks	72
Tuning iPlanet Directory Server	73
Improving the LDAP Read Performance	73
Tuning the Database Cache	73
Tuning the Entry Cache	74
Improving the LDAP Write Performance	74
Setting the db_home_directory	75
Distributing the Write Load Among Multiple Disks	76
Disabling Durable Transactions	77
Setting the ALL_IDS_THRESHOLD	78
LDAP Indexes	78
VLV Indexing	82
Tuning Oracle	82
Distributing the Tablespaces Among Multiple-Disks	82
Tuning the iPlanet Application Server	84
Creating New KJS Processes and Tuning	85
Session Model	86
Advantages	86
Disadvantages	86
Load Balancing	87
Setting the Heap Size for JVM	89
Tuning iPlanet Market Maker	89

Turning Off the Pricing Engine	89
Monitoring Performance	90
Monitoring Tools	90
Monitoring I/O Usage	90
Monitoring CPU Utilization	90
Monitoring Process Resource Utilization	90
Monitoring Multi-Processor CPU Usage	91
Observing Memory Usage	91
This gives a long full listing about every process with the processor number and memory utilization.	91
Monitoring Network Activity	91
Log Files	91
iPlanet Market Maker Logs	91
iPlanet Application Server Logs	92
iPlanet Directory Server Logs (LDAP)	93
Chapter 5 Verifying the Deployment	95
Deployment Verification Checklist	95
Recommended DBA Maintenance Activities	97
Chapter 6 Troubleshooting	101
Application server and web server would not start.	102
Display profile contained bad data.	102
E-mail notifications are not sent out.	103
kjs process crashes under load.	103
LDAP error occur before login.	103
Login fails for interoperability.	104
New users are not able to log in.	104
No available http sessions.	104
ORA-29861: domain index is marked LOADING/FAILED/UNUSABLE.	105
Oracle exceptions caused by Job Scheduler.	105
Oracle JDBC exceptions occur.	105
Page flow doesn't correspond with actions.	106
Slow login time.	107
Slow performance.	107
Slow response time.	107
Appendix A Oracle Table Sizing Information	109
Index	115

List of Figures

Figure 1-1	iPlanet Market Maker Architecture.	18
Figure 1-2	iPlanet Market Maker Business Process Flow.	19
Figure 1-3	Basic Production Configuration	33
Figure 1-4	SSL Production Configuration	35
Figure 1-5	High-Availability Production Configuration	37

List of Tables

Table 1	Chapter Summary	12
Table 1-1	Deployment Task Checklist	21
Table 1-2	Software Requirements for iPlanet Market Maker 4.5	23
Table 1-3	Hardware Recommended for Basic Production Setup	32
Table 1-4	Hardware Recommended for Production Setup With SSL	34
Table 1-5	Hardware Recommended for High Availability Production With SSL	36
Table 1-6	Community Users	38
Table 1-7	Details of the Production Setups	38
Table 2-1	Sizing Calculator	42
Table 2-2	Sizing Calculator Results for Example 1	44
Table 2-3	Sizing Calculator Results for Example 2	45
Table 2-4	Sizing Calculator Results for Example 3	46
Table 3-1	Oracle Initialization Parameters	48
Table 3-2	Oracle Settings	52
Table 3-3	Recommended Sizes of extents for Catalog Database	55
Table 3-4	Recommended Sizes of extents for Auctions Database	56
Table 3-5	Recommended Sizes of extents for Exchange Database	56
Table 3-6	Recommended Sizes of extents for In Box Notification	57
Table 4-1	LDAP Indexes	79
Table 5-1	Verification Checklist	95
Table A-1	Oracle Table Sizing Information	110

About This Document

This guide contains information and guidelines for deploying the 4.5 version of the iPlanet Market Maker software.

This preface contains the following sections:

- Audience
- What's in This Document
- Documentation Conventions
- The Document Online
- Related Documentation
- Other Useful Documentation
- Product Support

Audience

The audience for this guide includes iPlanet Professional Services, authorized system integrators, and any iPlanet or customer staff who are deploying or tuning the iPlanet Market Maker software.

What's in This Document

Table 1 summarizes what each chapter covers. The leftmost column describes subject matter that the chapter contains; the rightmost column identifies the chapter where you can find this information.

Table 1 Chapter Summary

If you want to know about this	See this chapter
Description of contents of this guide; listing of documentation set; information on product support	"About This Document"
How to plan for deployment of the iPlanet Market Maker system.	Chapter 1, "Planning the Deployment"
How to determine hardware needs.	Chapter 2, "Sizing Your System"
How to configure your system; how to import and configure data	Chapter 3, "Configuring Your System"
How to tune your system for optimum performance.	Chapter 4, "Performance Tuning and Monitoring"
How to verify the success of the deployment.	Chapter 5, "Verifying the Deployment"
Information and solutions on problems that may arise during upgrade	Chapter 6, "Troubleshooting"
Information on sizing Oracle tables.	Appendix A, "Oracle Table Sizing Information"

Documentation Conventions

This document uses the following conventions:

- The `monospace` font is used for sample code and code listings, Application Program Interface (API) and language elements (such as method names and property names), file names, path names, directory names, Hypertext Markup Language (HTML) tags, and any text that must be typed on the screen.
- The *italic* font is used in code to represent placeholder parameters (variables) that should be replaced with an actual value.
- Brackets ([]) are used to enclose optional parameters.
- A slash (/) is used to separate directories in a path.

The Document Online

You can find this document online in Portable Document Format (PDF) and HTML formats at the following web site:

<http://docs.iplanet.com/docs/manuals/marketmaker.html>

Related Documentation

NOTE Documentation for all iPlanet products can be found at the following web site:

<http://docs.iplanet.com/docs/manuals/>

The iPlanet Market Maker documentation set includes:

- *Release Notes*—Contains important information on the current release of iPlanet Market Maker. Read this document before working with the new iPlanet Market Maker release.
- *Installation Guide*—Provides instructions for installing the iPlanet Market Maker product and its enabling software.
- User interface Help (*Online Help*)—Provides guidelines and instructions for performing the iPlanet Market Maker tasks.
- *Catalog Import Guide*—Provides guidelines and instructions for setting up and running the Import utility.
- *Customization Guide*—Provides guidelines for customizing the existing iPlanet Market Maker user interface and module functionality.
- *Exchange Module Customization Guide*—Provides guidelines for customizing services to create exchanges.
- *Deployment Guide*—Provides guidelines for planning the deployment, installing and configuring the software, importing and configuring data, and tuning and monitoring performance.
- *Upgrade Guide*—Provides guidelines for planning, preparing and performing upgrades to the iPlanet Market Maker software.

Other Useful Documentation

- *iPlanet Application Server Installation Guide*—Provides guidelines for installing the iPlanet Application Server.
- *iPlanet Application Server Administrator's Guide*—Provides guidelines for tasks carried out by the administrator of one or more iPlanet Application Server machines.
- *iPlanet Web Server Administrator's Guide*—Provides guidelines for configuring, administering, and maintaining the iPlanet Web Server.
- *iPlanet Directory Server Documentation*—Provides guidelines for platform requirements and the iPlanet Directory Server installation.
- *ECXpert Administrator's Guide*—Explains concepts, structure, functions, and operation of iPlanet ECXpert, and provides guidelines for administering the system.
- *ECXpert Operations Reference Guide*—Explains ECXpert operational issues, and provides a reference on error messages that can be generated by ECXpert or passed through from third-party software components that ECXpert uses.
- *Oracle 8.1.7 / 9i Installation Guide for Sun SPARC Solaris*—Provides guidelines for installing Oracle.
- *Oracle 8.1.7 / 9i Designing and Tuning for Performance*—Provides guidelines for Oracle database design, and for tuning and monitoring performance.
- *Oracle 8.1.7 / 9i interMedia Text Reference*—Provides guidelines for text searching, retrieval, and viewing capabilities.
- *Actuate* documentation.

Product Support

If you have problems with your iPlanet Market Maker system, contact iPlanet customer support using one of the following mechanisms:

- Visit the iPlanet online support web site at:

<http://www.iplanet.com/support/online/>

From this location, the CaseTracker and CaseView tools are available for logging problems.

- Use the telephone dispatch number associated with your maintenance contract.

So that the technical support staff can best assist you in resolving problems, have the following information available when you contact support:

- Description of the problem, including the situation where the problem occurs and its impact on your operation.
- Machine type, operating system version, and product version, including any patches and other software that might be affecting the problem.
- Detailed steps on the methods you have used to reproduce the problem.
- Any error logs or core dumps.

Planning the Deployment

This chapter discusses planning the deployment of the iPlanet Market Maker software for your environment.

To accomplish full deployment of the iPlanet Market Maker software, you need to follow the instructions in the iPlanet Market Maker *Installation Guide*, *Customization Guide*, and possibly the *Upgrade Guide*.

This chapter contains the following sections:

- Overview of iPlanet Market Maker
- Overview of the Deployment Process
- Planning for Installation
- Software Requirements
- Developing the Deployment Plan
- Hardware Configurations
- Summary

Overview of iPlanet Market Maker

The iPlanet Market Maker software is a business-to-business e-commerce platform for building digital marketplaces where buying and selling companies can come together to conduct business online.

The iPlanet Market Maker software consists of a set of JSPs, Java servlets, and Java classes that are deployed on an iPlanet Application Server. The modular architecture of the software makes it an extensible platform to build custom marketplaces.

These modules are combined with a customizable portal layer to provide personalized views and content to participants. The platform is based on open standards such as J2EE technology, HTTP, LDAP, and XML. It is also made up of many other well-defined industry standard components and interfaces.

Figure 1-1 shows the architecture of the iPlanet Market Maker software.

Figure 1-1 iPlanet Market Maker Architecture.

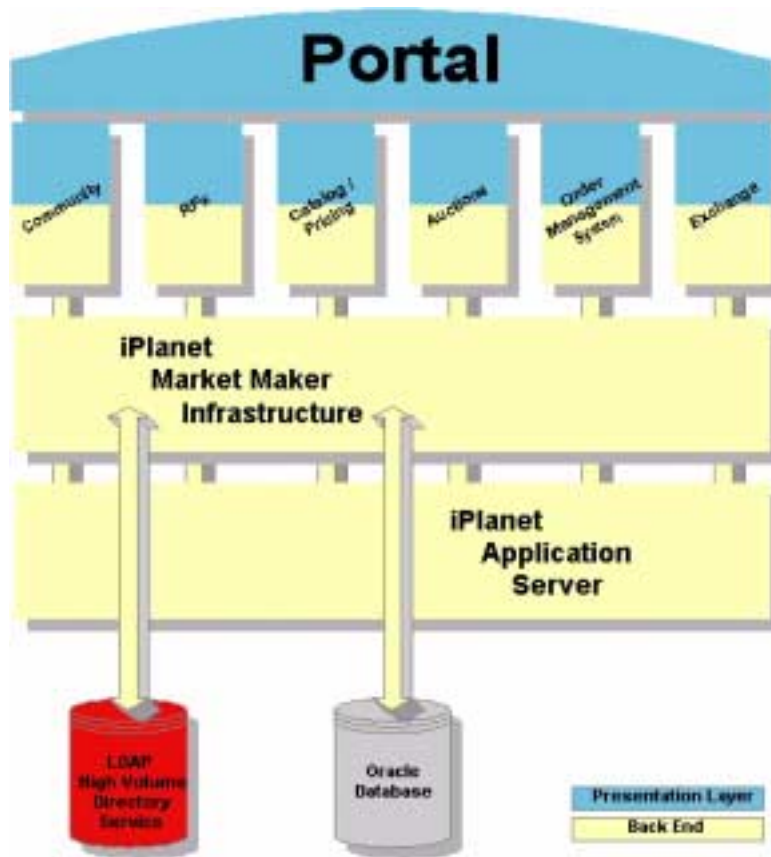
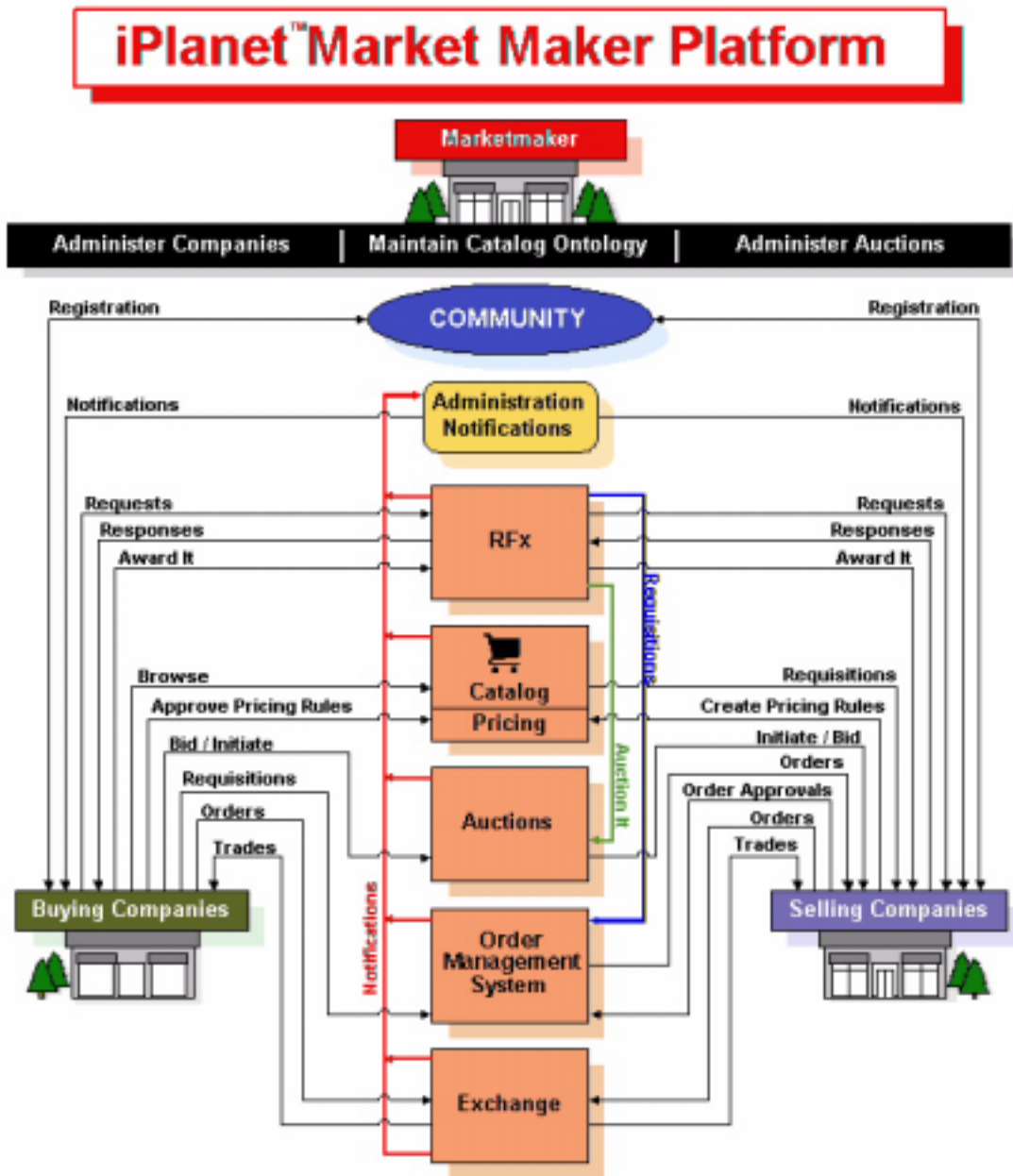


Figure 1-2 illustrates the business process flow in a production environment.

Figure 1-2 iPlanet Market Maker Business Process Flow.



Copyright © 2002 Sun Microsystems, Inc. All rights reserved.

Overview of the Deployment Process

iPlanet Market Maker is supported by a number of other products. The major steps for deploying iPlanet Market Maker are:

1. Installing and configuring hardware systems.
2. Installing and configuring software components, including:
 - o Solaris
 - o Oracle servers
 - o iPlanet Web Server
 - o iPlanet Application Server
 - o iPlanet Messaging Queue
 - o Actuate Reports
 - o iPlanet Directory Server (replicated)
 - o iPlanet Market Maker
3. Configuring the software components
4. Formatting and importing data, such as catalogs and price lists
5. Integrating the iPlanet Market Maker system with other applications and systems, such as iPlanet Messaging Queue and the iPlanet Portal Server
6. Tuning performance to make sure that the iPlanet Market Maker software is functioning smoothly and according to the unique needs of the customer
7. Verifying success of the deployment

For information on the known problems and limitations, refer to Chapter 6, “Troubleshooting.” Before installing the supporting software products, read this chapter so that you have a better understanding of any known issues and workarounds.

Table 1-1 provides a checklist of the high-level tasks for deploying the iPlanet Market Maker application.

- The left column indicates the sequence in which the task should be performed.
- The center column describes the task.

- The right column provides information about where to find instructions, guidelines, or tools for performing that task.

Table 1-1 Deployment Task Checklist

	Task	Location of Instructions/Guidelines/Tools
1	Plan the deployment—determine timeline, team members, locales, requirements	Chapter 1, “Planning the Deployment”
2	Determine hardware and sizing needs.	Chapter 2, “Sizing Your System” Sizing Calculator tool Developing the Deployment Plan section
3	Install hardware.	Documentation for the hardware.
4	Install software.	iPlanet Market Maker <i>Installation Guide</i> Software Requirements section Documentation for the software components
	A—Solaris operating system	Solaris documentation
	B—Oracle database server and client	<i>Installation Guide</i> , Oracle documentation
	C—iPlanet WebServer	<i>Installation Guide</i> for iPlanet WebServer
	D—Browser	<i>Installation Guide</i>
	E—iPlanet Application Server	<i>Installation Guide</i> for iPlanet Application Server
	F—iPlanet Messaging Queue	<i>Installation Guide</i> for iPlanet Messaging Queue
	G—iPlanet Directory Server	<i>Installation Guide</i> for iPlanet Directory Server
	H—iPlanet Portal Server	<i>Installation Guide</i> for iPlanet Portal Server
	I—Actuate Report package	<i>Installation Guide</i> , Actuate documentation
	J—iPlanet BuyerXpert	iPlanet Market Maker <i>Installation Guide</i>
5	Verify installation success.	
6	Configure software.	Chapter 3, “Configuring Your System”
	A—iPlanet WebServer	
	B—iPlanet Application Server	
	C—iPlanet Directory Server	
	D—Oracle	Chapter 3, “Configuring Your System” Oracle documentation
	E—iPlanet Market Maker	Chapter 3, “Configuring Your System”
	F—Actuate	Chapter 3, “Configuring Your System” iPlanet Market Maker <i>Installation Guide</i> Actuate documentation

Table 1-1 Deployment Task Checklist (*Continued*)

	Task	Location of Instructions/Guidelines/Tools
7	Verify configuration.	Chapter 3, “Configuring Your System”
8	Tune for performance and monitor system.	Chapter 4, “Performance Tuning and Monitoring”
9	Verify final deployment	Chapter 5, “Verifying the Deployment”
10	Solve any problems.	Chapter 6, “Troubleshooting”
11	Customize your system.	iPlanet Market Maker <i>Customization Guide</i>

Planning for Installation

During deployment planning, you should have decided on the production setup and the server architecture for your marketplace. Don’t start the installation process until your deployment planning is complete.

NOTE If you had iPlanet Market Maker version 1.0, 1.0 SP1, 4.0, or 4.0 SP1 installed, you need to upgrade to version 4.5. For details, refer to the *Upgrade Guide* for iPlanet Market Maker and the iPlanet Market Maker *Installation Guide*, version 4.5. If you do not have a production environment set up using the earlier versions of iPlanet Market Maker software, follow the steps below.

Further information on preparing, installing, and configuring the hardware and software systems is contained in the sections of this chapter. For detailed instructions on installing the iPlanet Market Maker software, refer to the iPlanet Market Maker *Installation Guide*.

The order of high-level installation steps is:

1. Ensure connection with the Oracle database.
2. Check the Java version. At the command prompt, type:

```
java -version
```

It should report the following:

```
1.2.2
```

3. Stop your iPlanet WebServer.

4. Verify that the iPlanet Directory Server is running.
5. Extract the iPlanet Market Maker installation files and install the software on the server(s) where the iPlanet Application Server is installed.
6. Install and configure Solaris (with appropriate patches) on the servers designated to host the Oracle database, iPlanet Directory Server, iPlanet Application Server, iPlanet WebServer, and the iPlanet Market Maker software.
7. Install and configure Oracle with the interMedia option, iPlanet Directory Server, iPlanet WebServer, and iPlanet Application Server on the designated servers.

NOTE If you need to install iPlanet Directory Server and iPlanet WebServer on the same machine, install iPlanet Directory Server first.

8. Install JDK patches for Solaris.
9. Install the JDK.

Software Requirements

The following software supports the iPlanet Market Maker 4.5 product:

Table 1-2 Software Requirements for iPlanet Market Maker 4.5

Component	Supporting Software	Comments
Platforms	Sun Solaris (SPARC) 2.6	Kernel patch 29 or higher, plus required iAS 6.5 patches
	Sun Solaris (SPARC) 8	Kernel patch 5 or higher, plus required iAS 6.5 patches
Browsers	Netscape Communicator, Version 4.7x, 6.2	Platforms: Solaris 2.6, 8, Windows 98, 2000, NT
	Microsoft Internet Explorer, Version 5.0, 6.0	Platforms: Windows 98, 2000, NT

Table 1-2 Software Requirements for iPlanet Market Maker 4.5

Component	Supporting Software	Comments
iPlanet Infrastructure Products	iPlanet WebServer (iWS), Version 4.1 SP9, 6.01	See the iPlanet Web Server (iWS) section in the Installation Guide.
	iPlanet Application Server (iAS), Version 6.5	See the Installing the iPlanet Application Server in the Installation Guide.
	iPlanet Directory Server, Version 4.13, 5.0 SP1	See the Installing the iPlanet Directory Server in the Installation Guide.
	iPlanet Messaging Queue, Version 2.0	See the Installing iPlanet Messaging Queue in the Installation guide.
Third Party Products	Oracle Server Database 8.1.7 / 9i Standard or Enterprise Edition, with InterMedia search engine	Customer must get Oracle license.
	Oracle Client 8.1.7 / 9i (if Market Maker is not installed on the same machine)	Customer must get Oracle license.
	*Actuate Report Server, Version 5.0	Bundled. NT workbench for customizing reports
Other	Java Developer's Kit (JDK), Production 1.2.2_07	Sun product
*Instructions for deploying reports are contained in the iPlanet Market Maker <i>Installation Guide</i> .		

Patches

Patches may be required for some of the supporting software products. Before installation, it is important to check the release notes for the supporting software to see if patches are required and how to get them.

iPlanet Patch Information

- iPlanet patches can be downloaded from the following web site:
<http://www.iplanet.com/downloads/patches/>
- Release notes for iPlanet products can be found at the following web site:
<http://docs.iplanet.com/docs/manuals/>

Solaris Patch Information

Refer to the iPlanet Market Maker *Installation Guide* for a list of the patches that are required or recommended for iPlanet Market Maker 4.5.

JDK 1.2.2_07 + JDK patches for Solaris are available at:

<http://java.sun.com/products/jdk/1.2>

In addition to the patches listed in the table, you may also want to install the latest patch cluster for your version of Solaris that includes additional recommended appropriate links starting at:

<http://sunsolve.sun.com>

Patches can be obtained from your service provider or can be downloaded from the web. To download the patches, start at the following location and follow the links to download the JDK (the patches are available on the same page as the JDK):

<http://www.sun.com/solaris/java>

For more information, refer to:

<http://www.sun.com/software/solaris/jre/download.html>

Developing the Deployment Plan

Before installing the iPlanet Market Maker software, the implementation team must develop a deployment plan to ensure optimal performance of the system. The deployment plan should identify the material contained in following section:

- Determining the Development Setup
- Determining the Marketmaker Requirements
- Estimating Initial Hardware Requirements
- Determining the Production Setup

Determining the Development Setup

To provide an environment for customizing and testing the iPlanet Market Maker software before sending it to the production environment, you should set up one of the following:

- Basic Development Setup

- Advanced Development Setup
- Basic Production Setup
- Production Setup with SSL
- High-Availability Production Setup

Basic Development Setup

This setup provides a platform for modifying the software and performing minimal testing after modifications are made. Typically, this setup consists of a single machine where all the software components (iPlanet Market Maker, iPlanet WebServer, iPlanet Application Server, iPlanet Messaging Queue, iPlanet Directory Server (LDAP) database, and Oracle database) are installed.

NOTE You need to install iPlanet Messaging Queue only if you are deploying the Exchange module.

Advanced Development Setup

This setup expands on the basic development setup by adding a quality assurance function to validate expected performance on a particular configuration. This setup accounts for the fact that user load will be distributed among several servers and that the database will run on separate machines.

The advanced setup consists of the following:

- At least one server where the iPlanet WebServer, iPlanet Application Server, and iPlanet Market Maker software are installed.
- At least one server where the iPlanet Messaging Queue, iPlanet Directory Server (LDAP) database, and Oracle database are installed.

NOTE You need to install iPlanet Messaging Queue only if you are deploying the Exchange module.

The advanced development setup is recommended over the basic development setup.

NOTE The advanced setup recommended specifies only the minimum requirements. However, it can be expanded to install the server applications on separate machines.

Determining the Marketmaker Requirements

To determine the requirements of the marketmaker organization you need to consider the following:

- **Availability**—Determine the requirements in terms of how important it is for the marketplace to remain up and running.
- **Security**—Determine the requirements in terms of the need for security features, such as SSL decryption.
- **Expected usage (user load)**—Estimate how heavily the different modules of the iPlanet Market Maker software will be used in terms of the following:
 - Number of concurrent users—The most common variable in determining the scalability of any e-commerce platform is the number of concurrent users that the platform needs to support.

NOTE *Concurrent users* are those actively running the software at one time, which is not necessarily equal to the number of users logged into the marketplace at one time.

- Intensity of usage—In order to size a production setup and tune an installation for better performance, the implementation team should quantify the intensity of usage for individual functions in terms of numbers of simultaneous users. The most commonly used functions, which were included in the mixed scenarios tested in our performance test labs, include the following:
 - Catalog—Users browsing and searching.
 - Auctions—Users bidding.
 - RFx—Buyers creating requests and sending the requests through distribution lists, and sellers creating responses.

- Order Management System—Buyers creating requisitions and submitting purchase orders for approval, and sellers approving or rejecting purchase orders.
- Community Registration—New companies and users registering.
- Exchange—Traders placing orders for commodities.
- Size of the database—Although the iPlanet Market Maker software is designed to work with large databases, it is normal to expect that a substantial increase in the size of a database could have an impact on the response times of the software. However, there is only a minimal relationship between the size of the database and performance of the software in terms of response times.

Estimating Initial Hardware Requirements

Based on the performance tests conducted using the development setup, you can develop a sizing calculator to derive an initial estimate of the size of the hardware configuration to meet the requirements of the marketmaker organization.

Refer to Chapter 2, “Sizing Your System,” for information on sizing.

Determining the Production Setup

Based on the policy considerations of the marketmaker organization and levels of security and availability required, you should select from the following production setups:

Basic Production Setup

The *basic production setup* includes facilities for balancing the load and connecting the web servers to a single port. This provides a level of security acceptable for marketmaker organizations that have separate enterprise security mechanisms or that run the software on intranets or virtual private networks (VPNs). If the enterprise security group at a marketmaker organization is not concerned with the security of data as it travels along the wire, this setup should be sufficient.

The basic production setup includes:

- Several web servers (iPlanet WebServer)
- Several application servers (iPlanet Application Server) in which the iPlanet Market Maker software is installed.

- A server for the iPlanet Directory Server (LDAP) database and iPlanet Messaging Queue.
- A server for the Oracle database.
- At least one firewall to protect the iPlanet Application Servers from the Internet

NOTE The basic production setup has been tested in our performance labs.

Production Setup with SSL

The *production setup with SSL* expands on the basic production setup by encrypting all communication sent over the Internet, ensuring that the data is secure. In addition to the elements used in the basic production setup, the production setup with SSL includes the following:

- More powerful servers at the web server layer, so that SSL decryption can be performed on the data before the data reaches the web servers.
- A separate web server for static content. This reduces the encryption overhead by removing static content that is not sensitive (for example, images).

NOTE The production setup with SSL has *not* been tested in our performance labs.

High-Availability Production Setup

The *high-availability production setup* expands on the SSL production setup by taking measures to ensure the highest levels of security and availability. In addition to the elements used in the production setup with SSL, the high-availability production setup includes elements such as the following:

- An Oracle Parallel Server for high availability of database using Sun Cluster technologies to ensure that the iPlanet Market Maker software can work in the event of any one Oracle instance going down. This is important when the Oracle database is part of a larger database containing critical data.
- Sun Cluster technologies to ensure that a hardware failure does not affect the marketplace.

- An iPlanet Directory Server using the Directory Server Replication configuration to replicate a directory server, and high availability. Alternatively, you can use the iPlanet Directory Access Router implementation, which offers a complete 7X24 solution for high availability of the directory. For more details, consult your sales representative.
- A message queue cluster.

NOTE The high availability production setup has *not* been tested in our performance labs.

Hardware Configurations

The iPlanet Market Maker software consists of a set of JSPs, Java servlets, and Java classes that are deployed on an iPlanet Application Server and static content on iPlanet WebServer. Consequently, when deploying the iPlanet Market Maker software, it is necessary to consider the scalability of the iPlanet Application Server, iPlanet WebServer, and the Oracle and LDAP databases

It is also important to achieve an optimized balance between the processing power assigned to the servers where the iPlanet Market Maker software is installed and the processing power assigned to the database servers. In all production setups, the iPlanet Market Maker software runs on servers that are separate from the database and directory servers where the data is stored.

NOTE The iPlanet Market Maker software was tested in the performance labs under the following conditions:

- A catalog containing no more than 100,000 items
- A community containing no more than 25,000 users

If these conditions are exceeded, you need to resize your hardware configuration to achieve better performance.

In the hardware configurations described in this section, we use mostly servers with four CPUs. Although you can use different machine configurations, you should consider the advantages of machine redundancy and machine similarity for load balancing and maintenance.

The components discussed here are:

- CPUs

- RAM
- Hard disk

Hardware Components

Each of these hardware components should be evaluated in relation to each other, so that the configuration of the machine is balanced. The purpose of balancing the right hardware component is to make sure that these components are utilized to the maximum extent without becoming CPU intensive or I/O intensive.

- CPU selection—To ensure scalability, you can select the latest versions of CPUs.
- RAM selection—RAM is a key component in machine performance. Because access to hard disks is thousands of times slower than memory access, enough RAM should be available to cache the data and prevent operating system paging. This would increase the system performance enormously. One GB of RAM for each processor is a safe estimate to start with, however, it could be lower or higher depending on the application being hosted. Apart from the application and the databases, operating system file systems, such as UFS, use RAM for UNIX file system buffer cache. Ensure that the highest density of RAM (2G RAM instead of two 1G RAM) is used so that spare RAM slots are available for upgrading RAM.
- Hard disk selection—Since I/O is the major factor influencing system performance, proper selection of drives, drive cache, and controllers is very important. Consider installing smaller and more drives instead of one large drive in order to achieve random I/O operation for databases. This can be achieved by using RAID options with multiple controllers. Selecting the drives with write-cache helps to improve physical I/Os by writing to the cache and avoiding I/O queues. When considering RAID, it is preferable that striping is done across controllers in hard drive storage arrays. Multiple controllers are required if the system is set up with any kind of mirroring.

For the best combination of hardware for different setups, consult iPlanet Professional Services for recommendations.

The following sections describe the hardware configurations for each of the three production setups described previously. Because the three examples are so similar with regard to hardware requirements, we present only one configuration for each setup. These configurations should be acceptable for both examples.

Basic Production Setup

In this case, the basic production setup for the companies would include a firewall layer of web servers to isolate the application servers running the application from the Internet.

Table 1-3 summarizes the hardware recommended for the basic production setup.

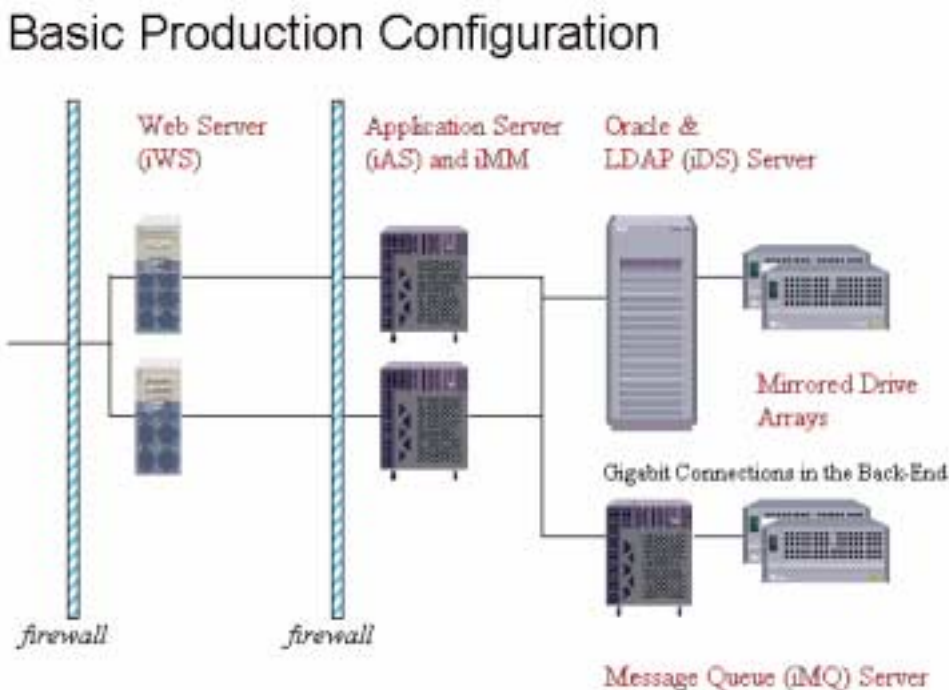
Table 1-3 Hardware Recommended for Basic Production Setup

Function	Machines	CPUs Per Machine	RAM Per Machine	Storage Per Machine
LDAP Server (iDS)	1 - Sun Enterprise 450	4 - 450 MHz/4MB Sun UltraSparcIIs/ UltraSparcIIIs	4 GB	2 StorEdge A5X00/A1000 4 X 18 GB
Oracle Server	1 - Sun Enterprise 6500	16 - 450 MHz/4MB Sun UltraSparcIIs/ UltraSparcIIIs	8 GB	2 StorEdge A5X00 24 x 9 GB
Application Server (iAS)	2 - Sun Enterprise 450s (can phase in more to accommodate additional load [concurrent user])	2 - 450 MHz/4 MB Sun UltraSparcIIs/ UltraSparcIIIs	4 GB	2 X 18 GB
Web Server (iWS)	2 - Sun Ultra 60s (can phase in more to accommodate additional load [concurrent user])	2 - 450 MHz/4 MB Sun UltraSparcIIs/ UltraSparcIIIs	1 GB	2 X 18 GB
Message Queue Server (iMQ)	1 - Sun Enterprise 450	4 - 450 MHz/4 MB Sun UltraSparcIIs/ UltraSparcIIIs	4 GB	2 X 18 GB

The *Basic Production Configuration* diagram (Figure 1-3) illustrates a basic production setup for the example companies. Although the sizing calculator indicated that the application server and web server layers would require four machines each, the diagram shows that you can start with two machines and phase

in additional machines as the user load increases. When phasing in additional machines, you should always add machines to the web server layer and application server layers in pairs, maintaining a 1:1 ratio between web servers and application servers to maintain proper fail-over.

Figure 1-3 Basic Production Configuration



Production Setup With SSL

In this case, the production setup with SSL for the companies would include a firewall layer, an upgraded layer of web servers to handle SSL decryption, and a separate web server for static content. Table 1-4 summarizes the hardware recommended for the production setup with SSL. Instead of upgraded web servers, it might be possible to use Sun's *Crypto Accelerator Boards* to off-load SSL processing from the CPU.

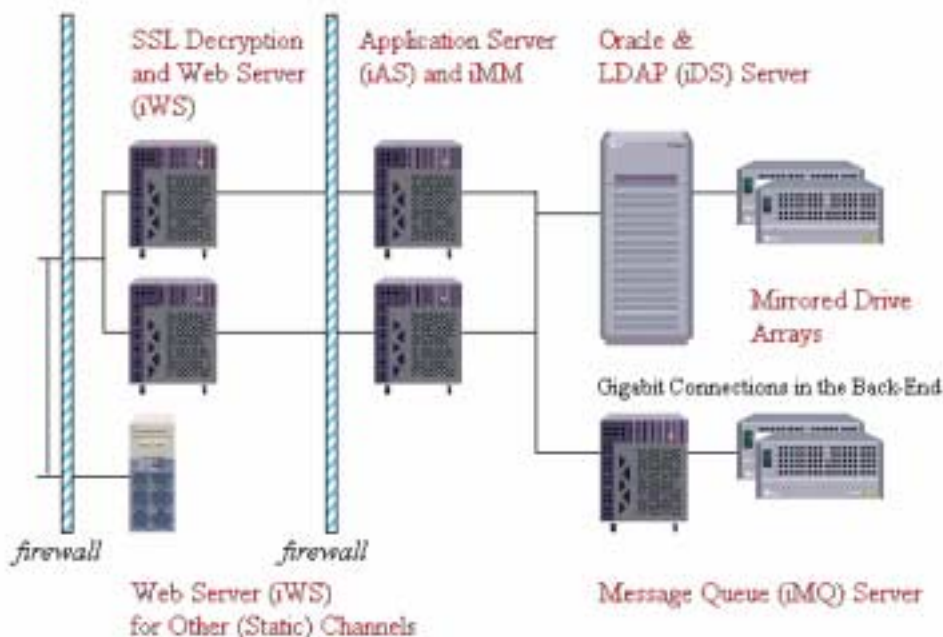
Table 1-4 Hardware Recommended for Production Setup With SSL

Function	Machines	CPUs Per Machine	RAM Per Machine	Storage Per Machine
LDAP Server (iDS)	1 - Sun Enterprise 450	4 - 450 MHz/4MB Sun UltraSparcIIs/ UltraSparcIIIs	4 GB	2 StorEdge A5X00/A1000 4 X 18 GB
Oracle Server	1 - Sun Enterprise 6500	16 - 450 MHz/4MB Sun UltraSparcIIs/ UltraSparcIIIs	8 GB	2 - StorEdge A5X00 12 x 18 GB
Application Server (iAS)	2 - Sun Enterprise 450s (can phase in more to accommodate additional load [concurrent user]) (Can use Sun Crypto Accelerator boards instead of switching from U60 to E450.)	4 - 450 MHz/4 MB Sun UltraSparcIIs/ UltraSparcIIIs	4 GB	2 X 18 GB
Web Server (iWS)	2 - Sun Enterprise 450s (can phase in more to accommodate additional load [concurrent user])	4 - 450 MHz/4 MB Sun UltraSparcIIs/ UltraSparcIIIs	4 GB	2 X 18 GB
Static Web Server	1 - Sun Ultra 60	2 - 450 MHz/4 MB Sun UltraSparcIIs/ UltraSparcIIIs	2 GB	2 X 18 GB
Message Queue Server (iMQ)	1 - Sun Enterprise 450	4 - 450 MHz/4 MB Sun UltraSparcIIs/ UltraSparcIIIs	4 GB	2 X 18 GB

The *SSL Production Configuration* diagram (Figure 1-4) illustrates a production setup with SSL for the example companies.

Figure 1-4 SSL Production Configuration

SSL Production Configuration



High-Availability Production Setup

In this case, the high-availability production setup for the companies would include a firewall layer, an upgraded layer of web servers to handle SSL decryption, and a separate web server for static content. The Oracle database should be hosted on a cluster, and the LDAP server should be a producer-consumer configuration. Table 1-5 summarizes the hardware recommended for the high-availability production setup.

The goal of this configuration is to have no single point of failure. Therefore, all of the following servers need a load balanced/clustered configuration:

- The Static Web Server
- The iAS-iMM Servers
- The iDS Server

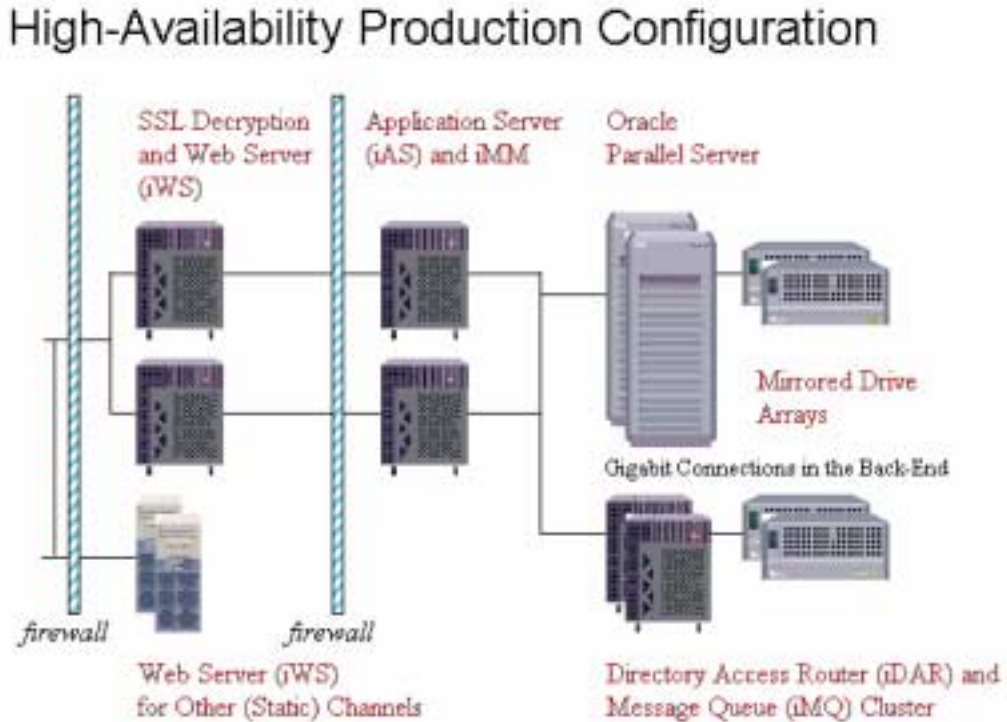
- The iMQ Server
- The Oracle Server

Table 1-5 Hardware Recommended for High Availability Production With SSL

Function	Machines	CPUs Per Machine	RAM Per Machine	Storage Per Machine
LDAP Server (iDS)	Directory Access Router Implementation	Please consult Professional Services for recommendations.		
Oracle Server	Oracle Parallel Server	Please consult Professional Services for recommendations		
Application Server (iAS)	2 - Sun Enterprise 450s (can phase in more to accommodate additional load [concurrent user])	4 - 450 MHz/4 MB Sun UltraSparcIIs/ UltraSparcIIIs	4 GB	2 X 18 GB
Web Server (iWS)	2 - Sun Enterprise 450s (can phase in more to accommodate additional load [concurrent user])	4 - 450 MHz/4 MB Sun UltraSparcIIs/ UltraSparcIIIs	4 GB	2 X 18 GB
Static Web Server	2 - Sun Ultra 60s + load balancing software	2 - 450 MHz/4 MB Sun UltraSparcIIs/ UltraSparcIIIs	2 GB	2 X 18 GB
Message Queue Cluster (iMQ)	2 - Sun Enterprise 450	4 - 450 MHz/4 MB Sun UltraSparcIIs/ UltraSparcIIIs	4 GB	2 X 18 GB

The *High-availability Production Configuration* (Figure 1-5) diagram illustrates a high-availability production setup for the example companies.

Figure 1-5 High-Availability Production Configuration



Summary

Table 1-6 and Table 1-7 below provide details of the development and production setups.

Note that the production setup with SSL and the high-availability production setup have not been through all of our performance tests.

Table 1-6 Community Users

	Basic Development Setup	Advanced Development Setup
Purpose	Application development and functionality validation. (Use all components running a single server.)	Application development and functionality validation. Validation performance with multiple web and application servers using a load-balancing configuration.
Hardware	1 - Sun Enterprise 450	2 - Sun Enterprise 450s for the application server 1 - Sun Enterprise 6500 for the database
Users	50	200
Catalog Items	4,000	16,000
Community Users	1,000	8,000

Table 1-7 Details of the Production Setups

	Basic Production Setup	Production Setup With SSL	High-availability Production Setup
Application	Use a web server layer and a layer of application servers.	Use an upgraded web layer, so that it can also handle SSL decryption, and a separate web server for static content.	The Oracle database should be hosted on a cluster, and the LDAP server should be configured for replication.

Table 1-7 Details of the Production Setups

	Basic Production Setup	Production Setup With SSL	High-availability Production Setup
Hardware	1 - Sun Enterprise 450 for the directory server 1 - Sun Enterprise 6500 for the database 4 - Sun Enterprise 450s for the application server 4 - Sun Ultra 60s for the web server	1 - Sun Enterprise 450 for the directory server 1 - Sun Enterprise 6500 for the database 4 - Sun Enterprise 450s for the application server 4 - Sun Enterprise 450s for the web server 1 - Sun Enterprise 450 for the static content	Directory Server Replication Configuration for the directory server Oracle cluster for the database 4 - Sun Enterprise 450s for the application server 4 - Sun Enterprise 450s for the web server 1 - Sun Enterprise 450 for the static content
Users	400	400	400
Catalog Items	1,000,000	1,000,000	1,000,000
Community Users	100,000	100,000	100,000

Summary

Sizing Your System

This chapter discusses how to use a sizing calculator to estimate the initial hardware requirements for your iPlanet Market Maker system.

This chapter contains the following sections:

- Overview of the Sizing Calculator
- Example 1: Auction-Intensive Marketplace
- Example 2: RFX-Intensive Marketplace
- Example 3: Exchange Intensive Marketplace

Overview of the Sizing Calculator

Based on the performance tests conducted using the development setup, you can develop a sizing calculator to derive an initial estimate of the size of the hardware configuration needed to meet the requirements of the marketmaker organization.

NOTE The calculator as discussed in this chapter is intended to be a rough guide. Additional parameters should be considered when estimating hardware requirements.

Table 2-1 represents a typical sizing calculator.

Table 2-1 Sizing Calculator

Module	Enter the Number of Concurrent Users Below	Response Time (Seconds)	Number of Pages Served Per Hour	iWS	Estimated Number of CPUs			
					iAS	LDAP	Oracle	iMQ
Community Registration								
Catalog Searching								
RFx Requests/Responses								
Auction Bidding								
Order Management Sys (OMS)								
Exchange								
Sub Total								
Total Number of CPUs Required With Safety Factor 1.5								

In the second column (Enter the Number of Concurrent Users Below) of the sizing calculator, enter the number of users that you expect to be actively and simultaneously performing each of the commonly used functions, as listed in the first column.

After entering the numbers, click Calculate to display the following results:

- **Response Time**—The number of seconds it takes for the software to respond to user requests submitted in each module.
- **Number of Pages Served per Hour**—The maximum number of actual pages served in an hour when the specified function is performed by the given number of users.
- **CPUs for iPlanet WebServer**—The following hardware requirements for the web server:
 - The number of CPUs required for the iPlanet WebServer for each functional module.
 - The total number of CPUs required for the iPlanet WebServer for all modules.

- The total number of CPUs required for the iPlanet WebServer for all modules, considering a reasonable load factor.
- CPUs for iPlanet Application Server—This column provides the following hardware requirements for the application server:
 - The number of CPUs required for the iPlanet Application Server for each functional module.
 - The total number of CPUs required for the iPlanet Application Server for all modules
 - The total number of CPUs required for the iPlanet Application Server for all modules, considering a reasonable load factor.
- CPUs for iPlanet Directory Server—This column provides the following hardware requirements for the LDAP database:
 - The number of CPUs required for the LDAP database for each functional module.
 - The total number of CPUs required for the LDAP database for all modules.
 - The total number of CPUs required for the LDAP database for all modules, considering a reasonable load factor.
- CPUs for Oracle—This column provides the following hardware requirements for the Oracle database:
 - The number of CPUs required for the Oracle database for each functional module.
 - The total number of CPUs required for the Oracle database for all modules.
 - The total number of CPUs required for the Oracle database for all modules, considering a reasonable load factor.
- CPUs for iPlanet Messaging Queue—This column provides the following hardware requirements to run iPlanet Messaging Queue:
 - The number of CPUs required to run the iPlanet Messaging Queue for the Exchange module.
 - The total number of CPUs required to run the iPlanet Messaging Queue for the Exchange module.
 - The total number of CPUs required to run the iPlanet Messaging Queue for the Exchange module, considering a reasonable load factor.

From the Total with Load Factor (x 1.5) values in the table, you can derive an initial estimate of hardware requirements for the iPlanet WebServer, the iPlanet Directory Server, and the Oracle database.

Example 1: Auction-Intensive Marketplace

Table 2-2 represents the deployment of the iPlanet Market Maker software by a marketmaker firm that plans to use the marketplace primarily for hosting auctions. If you enter the numbers shown below in the second column (Enter the Number of Concurrent Users Below) and click Calculate, you see the numbers shown in the table.

Table 2-2 Sizing Calculator Results for Example 1

Module	Enter the Number of Concurrent Users Below	Response Time (Seconds)	Number of Pages Served Per Hour	iWS	Estimated Number of CPUs			
					iAS	LDAP	Oracle	iMQ
Community Registration	10	3.983	3394.9	0.0064	0.2136	0.0480	0.1920	
Catalog Searching	50	4.096	18365	0.134	1.8660	0.12	1.8800	
RFx Requests/Responses	0	2.413	0	0	0	0	0	
Auction Bidding	100	1.933	29957	0.44	2.32	0.2133	2.1867	
Order Management Sys (OMS)	40	4.518	11010	0.04	0.8800	0.4640	0.944	
Exchange	0	1.235	0	0	0	0	0	0
Sub Total	200		62726.9	0.6204	5.2796	0.8453	5.2027	0
Total Number of CPUs Required With Safety Factor 1.5				0.9966	8.0394	0.6020	7.6480	0
							5	

From Example 1, you can estimate the hardware requirements as follows:

- iPlanet WebServer (iWS) = 1 CPU
- iPlanet Application Server (iAS) = 8 CPUs
- iPlanet Directory Server (LDAP) = 1 CPU
- Oracle = 8 CPUs

Example 2: RFx-Intensive Marketplace

Table 2-3 represents the deployment of the iPlanet Market Maker software by a marketmaker firm that plans to use the marketplace primarily for procurement (RFx). If you enter the numbers shown below in the second (Enter the Number of Concurrent Users Below) column and click Calculate on the calculator, you see the numbers shown in the table.

Table 2-3 Sizing Calculator Results for Example 2

Module	Enter the Number of Concurrent Users Below	Response Time (Seconds)	Number of Pages Served Per Hour	iWS	Estimated Number of CPUs			
					iAS	LDAP	Oracle	iMQ
Community Registration	10	3.983	3394.9	0.0064	0.2136	0.0480	0.1920	
Catalog Searching	50	4.096	18365	0.134	1.8660	0.12	1.8800	
RFx Requests/Responses	100	2.413	22407	0.1128	5.4872	1.28	7.5200	
Auction Bidding	0	1.933	0	0	0	0	0	
Order Management Sys (OMS))	40	4.518	11010	0.04	0.8800	0.4640	0.944	
Exchange	0	1.235	0	0	0	0	0	0
Sub Total	200		55176.9	0.2932	8.4468	1.912	10.536	0
						0		
Total Number of CPUs Required With Safety Factor 1.5				0.5058	12.7902	2.202	15.648	0

From Example 2, you can estimate the hardware requirements as follows:

- iPlanet WebServer (iWS) = 1 CPU
- iPlanet Application Server (iAS) = 13 CPUs
- iPlanet Directory Server (LDAP) = 3 CPU
- Oracle = 16 CPUs

Comparing Example 2 with Example 1, you can see that the total number of simultaneous users is the same. However, because the RFx module requires more processing power than the Auctions module, the number of CPUs required in Example 2 is greater.

Example 3: Exchange Intensive Marketplace

Table 2-4 represents the deployment of the iPlanet Market Maker software by a marketmaker firm that plans to use the marketplace primarily for hosting exchanges. If you enter the numbers shown below in the second (Enter the Number of Concurrent Users Below) column and click Calculate on the calculator, you see the numbers shown in the table.

Table 2-4 Sizing Calculator Results for Example 3

Module	Enter the Number of Concurrent Users Below	Response Time (Seconds)	Number of Pages Served Per Hour	iWS	Estimated Number of CPUs			
					iAS	LDAP	Oracle	iMQ
Community Registration	10	3.983	3394.9	0.0064	0.2136	0.0480	0.1920	
Catalog Searching	50	4.096	18365	0.134	1.8660	0.12	1.8800	
RFx Requests/Responses	0	2.413	0	0	0	0	0	
Auction Bidding	0	1.933	0	0	0	0	0	
Order Management Sys (OMS)	40	4.518	11010	0.04	0.8800	0.4640	0.944	
Exchange	100	1.235	30573	0.052	3.948	0.0864	1.28	2.08
Sub Total	200		63342.9	0.2324	6.9076	0.7184	4.296	2.08
Total Number of CPUs Required With Safety Factor 1.5				0.3366	4.5594	0.282	4.368	3.12

From Example 3, you can estimate the hardware requirements as follows:

- iPlanet WebServer (iWS) = 1 CPU
- iPlanet Application Server (iAS) = 5 CPUs
- iPlanet Directory Server (LDAP) = 1 CPU
- Oracle = 5 CPUs
- iPlanet Messaging Queue = 3 CPUs

Comparing this example with Examples 1 and 2, you can see that the total number of simultaneous users is the same. However, because the RFx and Auctions modules each require more processing power than the Exchange module, the number of CPUs required in Example 1 and in Example 2 is greater.

Configuring Your System

This chapter contains the following sections:

- Configuring Oracle
- Configuring iPlanet Market Maker for the Databases
- Importing and Configuring Data

Configuring Oracle

The default installation is sized for a small sample database. The Oracle database should be set up and configured to meet the specific requirements of your marketplace. Some recommendations on configuring Oracle are provided in this section. For more details, refer to the document, Oracle 8.1.7 / 9i Designing and Tuning for Performance.

Setting Oracle Initialization Parameters

The following are the recommended changes to the Oracle initialization parameters in the `init<SID>.ora` file.

1. Resize the System Global Area (SGA).

The SGA is the memory area that is available to an iPlanet Market Maker installation. To achieve optimal performance, it is recommended to set the SGA size as large as possible, depending on the memory available on the server. This can be achieved by changing the values for shared SQL pool and data block buffer cache settings.

- a. Set shared SQL pool—The value for this setting can be changed by setting `shared_pool_size` parameter (in bytes). Start by setting this value to 400 MB, and monitor the usage of shared SQL pool.
 - b. Set data block buffer cache size—The data block buffer cache size is set by the `db_block_buffers` `init.ora` parameter (in database blocks). The recommended size of data block buffer cache is 2 percent of the database size. The recommended `db_block_size` for the iPlanet Market Maker installation is 8192 bytes. Start by setting this value to 96000 and monitor the cache hit ratio.
2. Improve sorting operation—Set the `sort_area_size` and `sort_area_retained_size` to minimize the sorts (disk) to less than 5% of the total sorts, because searching the catalog is resource intensive due to the hierarchical nature of queries and the interMedia option.
 3. Set the `db_block_lru_latches` parameter to a number equal to the number of CPUs on a multiple-CPU system.

Oracle Initialization Parameters

Performance tests were conducted on an iPlanet Market Maker installation for the functions:

- Catalog loading
- Catalog searching
- Catalog browsing
- Auction bidding
- Community registration (companies and users).

The installation was sized to handle 500 concurrent users and a 2 GB database. The following parameter settings are recommended based on the results of the above test

Table 3-1 Oracle Initialization Parameters

Name	Value
Oracle Version compatible	8.1.0
<code>db_block_buffers</code>	96000
<code>db_block_lru_latches</code>	30
<code>db_block_size</code>	8192

Table 3-1 Oracle Initialization Parameters (*Continued*)

Name	Value
instance_name	VORTEX
enqueue_resources	20000
log_buffer	5120000
open_cursors	500
rollback_segments	(RBS0, RBS1, RBS2, RBS3, RBS4, RBS5, RBS6, RBS7, RBS8, RBS9, RBS10, RBS11, RBS12, RBS13, RBS14, RBS15, RBS16, RBS17, RBS18, RBS19, RBS20, RBS21, RBS22, RBS23, RBS24, RBS25, RBS26, RBS27, RBS28)
shared_pool_size	419430400
sort_area_retained_size	1024000
sort_area_size	5120000

For an explanation of other parameters, refer to the Oracle 8.1.7 / 9i *Administrators Guide*.

Configuring Net8

All the installation scripts use SQLPLUS, which in turn connects using the Net8 configuration to support installation from client machines. If you are installing on the server machine, make sure that Net8 is configured.

SQLPLUS should be able to connect to the net_service_name setup in Net8. For example

```
sqlplus userid/passwd@net_service_name
```

Configuring iPlanet Market Maker for the Databases

After the installation, the iPlanet Market Maker software needs to be configured to ensure that the connections to the LDAP and the Oracle databases are available from the iPlanet Market Maker host and that the job scheduler runs properly.

The following sections discuss the tasks involved:

- Connecting to Remote LDAP Instances
- Connecting to Remote Oracle Instances
- Configuring the Schemas
- Sizing Objects (Tables and Indexes)
- Setting the Job Scheduler

Connecting to Remote LDAP Instances

Presently the iPlanet Market Maker Installer is not equipped to carry out a remote login and installation of the LDAP schema and data from a web server/iPlanet Market Maker host. However, the Installer queries the user for the remote host names, port numbers, user name, and password of the LDAP, even while carrying out the installation of the iPlanet Market Maker core, in order to make the appropriate settings in the property files. Specifying the correct locations of these hosts and port numbers during installation of the iPlanet Market Maker software makes the appropriate changes to the `VortexConfiguration` and `userxConfiguration` files. Nonetheless, the following settings should be checked to ensure that they point to correct locations after the installation.

To connect to a remote LDAP instance

On the iPlanet Application Server/iPlanet Market Maker hosts, in the `userxconfig.properties` file, change the following settings to point to the correct host and port locations:

```
ldaphost: <remote host name>
ldapport: <port name on which the ldap instance listens >
```

On the web server/iPlanet Market Maker hosts, LDAP read and write user name and passwords can be found in the `VortexConfiguration.properties` file.

LDAP Connection Pool Settings

On the iPlanet Application Server/iPlanet Market Maker hosts in the `userxconfig.properties` file, change the following settings to new values:

```
ldappool.minimum=150 (Default Value : 10)
ldappool.maximum=300 (Default Value : 20)
```

Each connection handles client requests serially. The minimum and maximum settings depend on the number of concurrent users. 150/300 per LDAP connection pool is a safe setting for an expected number of 300 concurrent users per web server.

Connecting to Remote Oracle Instances

The setting that points to the database URL for the type four drivers needs to be changed to ensure that the remote host name is included in the URL instead of the local host name.

On the iPlanet Application Server/iPlanet Market Maker hosts in the `VortexConfiguration.properties` file, change the following setting to point to the correct host and port locations:

```
CFG_DATABASE_URL=-jdbc:oracle:thin:@<remote host
name>:1521:<instance_name>
```

Oracle Connection Pool Settings

On the web server/iPlanet Market Maker hosts in the `VortexConfiguration.properties` file, change the following settings to new values:

```
CFG_CPOOL_MAXIMUM=500 ( Default Value : 30)
```

Multiple database connections per user are required to be opened for some of the requests received from the code. To avoid running out of database connections, 500 is a safe value for the maximum connection pool size/web server for loads of up to 300 concurrent users through each iPlanet Market Maker host.

Configuring the Schemas

Each module (Catalog, Pricing, Community, Architecture (also known as Common), Auction, RFX, Exchange, and Order Management System) has its own database schema owner. The schema owners are CAT, PRI, COM, CMN, AUC, RFX, OMS and EXC. The passwords are entered during installation for each module (for example, CAT/CAT_PWD). All the objects (that is, tables, sequences, and so on) owned by these schema owners are granted to the application user VORTEX. The password for VORTEX user is entered during installation. The iPlanet Market Maker user interface connects to the database from the application servers using the application user (VORTEX).

Table 3-2 lists the values that need to be given for the Oracle settings during installation.

Table 3-2 Oracle Settings

Module Name	Schema / Password	Default Tablespaces		
		Data	Index	Temporary
Architecture (common)	CMN/CMN_PWD	CMN_TS	CMN_XTS	TEMP
Catalog	CAT/CAT_PWD	CAT_TS	CAT_XTS	TEMP
Pricing	PRI/PRI_PWD	PRI_TS	PRI_XTS	TEMP
Auction	AUC/AUC_PWD	AUC_TS	AUC_XTS	TEMP
Negotiations	RFX/RFX_PWD	RFX_TS	RFX_XTS	TEMP
Order Management System	OMS/OMS_PWD	OMS_TS	OMS_XTS	TEMP
Exchange	EXC/EXC_PWD	EXC_TS	EXC_XTS	TEMP
iPlanet Market Maker	VORTEX/VORTEX_PWD	CMN_TS	CMN_XTS	TEMP

Oracle schema creation is done as a part of installation using the iPlanet Market Maker Installer. The iPlanet Market Maker Installer provides the option of installing the following independently:

- iPlanet Market Maker core
- iPlanet Directory Server database
- Oracle database

Make sure that the option to create Oracle schema is checked (check box). The Installer uses the `$IMM_HOME/setup/bin/installsql.sh` script by passing the appropriate parameters. This install script uses the file `$IMM_HOME/setup/etc/backendmodules` which has a list of modules to be installed in the right order for the database schema. The module names correspond to the directories under `$IMM_HOME`. The main installation script calls sql install script `$IMM_HOME/<module>B/sql/install.sh`. These module specific install scripts create tablespaces, users, and all the database schema objects owner using the `$IMM_HOME/<module>B/sql/installParams.env` file. All the SQL scripts that create the schema have drop statements before creating the objects.

The following points are worth noting during schema creation:

- All the data table spaces are created in the directory, which is provided as “datafile location/datafile name” given during installation. The recommended naming convention for the data files is:

```
<abbreviated_name_of_module>01_Oracle_SID
```

An example for the Auctions module:

```
auc01_vortex
```

- All the index tablespaces are created in the directory which is provided as “index datafile location/datafile name” with the same data file size during installation.
- All the schema and application users are created, and their temporary tablespace set to use TEMP tablespace.
- The initial installation uses the default settings for storage parameters provided by the scripts for create table and index statements.
- The SQL scripts have a drop statement before each of the create statements. During initial installation, all the drop statements would throw errors, because there are no iPlanet Market Maker objects in the database. Subsequent runs will drop and recreate the tablespaces, users, and objects.

CAUTION Running the `$IMM_HOME/setup/bin/installsql.sh` script, or any SQL script, erases the iPlanet Market Maker schema without warning. Do not run these scripts in a production system when the iPlanet Market Maker installation is running and has data in it. These are install scripts and should be run only during installation.

- An execution log is generated each time the script runs. Also, a post install log file, which contains all the database script logs including the script execution and errors, is created. These logs can be helpful in trouble shooting. The script stops if it encounters any SQL statement errors due to create statements or connect statements. Otherwise, it will continue the execution even though there are other errors due to drop statements; these should be ignored.
- During installation, and before running the scripts, the Installer unzips and copies all the scripts to their appropriate locations. To make the changes to the schema scripts for storage and others, the scripts need to be edited and run again.

NOTE Make a backup of the file and do not change the name of the file.

Sizing Objects (Tables and Indexes)

It is recommended that the deployment team estimate the storage requirements for the modules and modify the storage parameters (hard disk space) of the tables and indexes for the production installation.

The installation scripts for iPlanet Market Maker software can be found in the `$IMM_HOME/setup/sql` and `$IMM_HOME/<module_name>/sql` directories. The directory `$IMM_HOME/setup/sql/schema` contains the create SQL script which creates the tablespaces and schema users. The Installer passes the datafile, index file location, and its sizes to this script. Also, each module (for example, cmn, auc, catalog and so on) has a directory sql, which contains the create table scripts (for example: `$IMM_HOME/setup/catalog/db/sql/schema/*.sql` files). Each table is created using these scripts. The storage parameters for the create tables scripts need to be examined, and changed based on the size of the marketplace. Default values are given to most of the create table parameters such as `pctfree`, `pctused`, `initial extents`, `next`, and `pctincrease`.

Database administrators are advised to change the sizing parameters for the tables depending on the data storage requirements for the tables. Appendix A, “Oracle Table Sizing Information” serves as a guide in calculating the storage parameters. The average row lengths presented in the table are based on performance tests carried out on iPlanet Market Maker installation in the test labs. The actual row lengths for each installation will vary, so the DBA should add 10 to 30 percent to the provided values based on their judgement.

There are two ways to recreate the schema using the changed parameters:

Re-installing the Database Schema Using the Installer

1. Run the iPlanet Market Maker Installer.
2. Uncheck the iPlanet Market Maker Core and iPlanet Market Maker Directory Server database.
3. Ensure that the iPlanet Market Maker Oracle database is checked.
4. Re-enter the same parameters that have been used during initial installation for encryption, passwords, LDAP details, and so on. Do not make any changes from initial input when re-installing.
5. Verify changes in the database.

Re-installing the Database Schema Using Scripts

1. Follow the installation instructions in the “Importing and Configuring Data” section.

2. Make sure that the VORTEX_SID environment parameter is set to net_service_name.
3. Verify that the module passwords (for example, auc_pwd) are set in the environment before running the script.
4. Verify changes in the database.

CAUTION The iPlanet Market Maker software does not support any changes to the database schema. No addition, modification, or deletion should be made to tables, columns, or indexes. The iPlanet Market Maker software does not support the use of SQLPLUS or any third party tools to manipulate the data directly. The iPlanet Market Maker APIs should be used to manipulate the data in accordance with the customization guidelines. For details, refer to the iPlanet Market Maker *Customization Guide*.

Catalog Module

The size of the catalogs (both the master catalog and the buyer catalogs) affects the response times of the search and browse operations directly. When the master catalog is loaded, some of the main tables opened are CAT_MASTER_ENTRY, CAT_ATTR, and CAT_SEARCH_ATTR. The number of initial and next 'extents should be configured to match the size of the catalog to achieve best performance. When buyers log in and create buyer catalogs (catalog views and not Oracle views), the table CAT_VIEW_ENTRY is populated with the pointers to the tables CAT_MASTER_ENTRY and CAT_ATTR. A seller catalog of 100K entries in CAT_MASTER_ENTRY can have 500K rows in CAT_VIEW_ENTRY, if five views are created by five different buyers (showing all the master catalog). So the number of rows in CAT_VIEW_ENTRY can grow as buyers create views of master catalogs. Creation of buyer catalogs also adds records into the CAT_ATTR table for each category.

The following table gives the recommended sizes of extents for core tables and their indexes for the Catalog module. These numbers can vary depending on catalog structure (categories and items) and the number of attributes for each item.

Table 3-3 Recommended Sizes of extents for Catalog Database

Oracle Table (Indexes)	Rows (Catalog items)	Recommended storage Size of extents
CAT_MASTER_ENTRY	100000	20 MB
CAT_MASTER_ENTRY(index)	100000	20 MB
CAT_VIEW_ENTRY	275000	60 MB

Table 3-3 Recommended Sizes of extents for Catalog Database (*Continued*)

Oracle Table (Indexes)	Rows (Catalog items)	Recommended storage Size of extents
CAT_VIEW_ENTRY(index)	275000	60 MB
CAT_ATTR	800000	130 MB
CAT_ATTR(index)	800000	130 MB
CAT_SEARCH_ATTR	20000	3 MB
CAT_SEARCH_ATTR(index)	20000	3 MB

Auctions Module

The size of extents for the core tables and indexes of the Auctions module is determined by the number of auctions and the number of bids for each auction

Table 3-4 Recommended Sizes of extents for Auctions Database

Oracle Table (Indexes)	Rows (Auction components)	Recommended storage size of extents
AUC_AUCTION	30000	50 MB
AUC_AUCTION(indexes)	30000	25 MB
AUC_BID_HISTORY	400000	100 MB
AUC_BID_HISTORY(indexes)	400000	60 MB

Exchange Module

The size of extents for the core tables and indexes of the Exchange module is determined by the number of orders and the number of trades.

Table 3-5 Recommended Sizes of extents for Exchange Database

Oracle Table (Indexes)	Rows (Exchange components)	Recommended storage size of extents
EXC_PROCESSED_MESSAGES	100000	30 MB
EXC_NVP_PASSIVE_ORDER	100000	30 MB
EXC_NVP_PASSIVE_AXIS	300000	60 MB
EXC_ORDER_REQUESTS	100000	40 MB

Table 3-5 Recommended Sizes of extents for Exchange Database (*Continued*)

Oracle Table (Indexes)	Rows (Exchange components)	Recommended storage size of extents
EXC_ORDERS_TEMP	100000	40 MB
EXC_ORDER_REQUEST_CRITERIA	1000000	40 MB
EXC_ORDER_REQUEST_LIFESPAN	100000	25 MB
EXC_ORDER_SCHEDULER	1000000	20 MB
EXC_ORDERS	100000	40 MB
EXC_ORDER_AXES	300000	80 MB

In-Box Notification

In-box notifications and messages are stored in the iPlanet Market Maker infrastructure under the tables INBOX_ENTRY and CMN_PLM. The INBOX_ENTRY table contains all the inbox notifications. CMN_PLM is the container for all the custom messages for each object based on Java property files.

Table 3-6 Recommended Sizes of extents for In Box Notification

Oracle Table (Indexes)	Rows (Auction components)	Recommended storage size of extents
INBOX_ENTRY	100000	60 MB
INBOX_ENTRY(indexes)	100000	25 MB
CMN_PLM	540000	45 MB
CMN_PLM(indexes)	540000	12 MB
CMN_PLM_STRING	650000	30 MB
CMN_PLM_STRING(indexes)	650000	14 MB

The extent sizes are calculated based on default sizing parameters (percent free, and so on) for the respective tables, as provided in the iPlanet Market Maker table scripts. If the values for other parameters are changed, the extent sizes should be recalculated.

Setting the Job Scheduler

The job scheduler runs in its own JVM, separate from the JVM in which the presentation and runtime layers of the iPlanet Market Maker software run.

The job scheduler process reads the loaded properties from the `VortexConfiguration.properties` file. This file has to be included in the classpath for the job scheduler process. The following properties must be modified for the job scheduler process:

- Maximum Heap Size for the JV
- OnceOnly and Repeatable Timer Thread Pool Sizes
- OnceOnly Worker Thread Pool Size
- Repeatable Worker Thread Pool Size
- Sleep Interval for OnceOnly Thread
- Repeat Sleep Interval
- Transactional Mail Scheduler Interval
- Max Threads in Notifier PoolM

Maximum Heap Size for the JV

To set the maximum heap size for the JVM in which the job scheduler process runs, set the `Xmx256M` option to 256 MB. This maximum heap size is sufficient for about 600 jobs registered with the job scheduler process.

```
#CFG_JS_EXE
# - The JobScheduler process.

37=java -Djava.compiler=NONE -Dvortex.logFile=jobscheduler.log
-Xmx256m

com.iplanet.ecommerce.vortex.util.scheduler.JobSchedulerProcess
```

OnceOnly and Repeatable Timer Thread Pool Sizes

Notifications about marketplace events, such as bidding on auctions, submitting orders to vendors, or approval of requisitions, are sent to users by email or to the users Shared In-Box. Users can configure this notification job, and the job scheduler is the process that ensures that such notification jobs are triggered.

The job scheduler process has a `OnceOnly` timer thread and a `Repeatable` timer thread.

- OnceOnly timers are related to jobs that have to be run only once at a pre-specified time (for example, Auctions opening and closing).
- Repeatable timers are related to jobs that must be run repeatedly at a pre-defined time interval.

These two threads keep a Job List updated for two worker thread pools:

- OnceOnly worker thread pool
- Repeatable worker thread pool

To provide for concurrent job requests, the numbers of workers in these two worker thread pools must be set to appropriate values.

OnceOnly Worker Thread Pool Size

The major load on the OnceOnly worker threads will be auctions opening and closing events. The actual value of the pool size should be determined based on the volume of auctions opening/closing on a site each day. For a fairly auction-intensive site, the safe value is 20, because all 20 worker threads will be called from their wait state only if there are 20 concurrent open auction or close auction jobs at the same instant on the site.

```
# CFG_JS_ONCEONLY_NUM_WORKERS
# Total number of workers in the thread pool for the OnceOnly
Timer.
38=20 ( Default value = 3 )
```

Repeatable Worker Thread Pool Size

The two repeatable timers that can be registered with the job scheduler are the Transactional Mail Scheduler and the RFx job scheduler.

- The Transactional Mail Scheduler is a repeatable timer implemented for reading new entries in mail_info and handing over the registered notifications to the Notifier thread pool for delivery to the Mail Transfer Agent.
- The RFx job scheduler is the RFx reaper that marks as pending those RFx requests and replies that are due to expire in less than or equal to one day.

To account for any concurrent repeatable requests from these timers, 4 is a safe value for the Repeatable thread pool size.

```
#CFG_JS_REPEAT_NUM_WORKERS
# - Total number of workers in the thread pool for the Repeat
Timer.
```

```
39=4 ( Default value = 3 )
```

Sleep Interval for OnceOnly Thread

The OnceOnly database polling thread polls the `cmn_job_scheduler_entry` table in order to read any new or revised timers after the last read in order to keep the job queue for the OnceOnly thread pool in synchronization with additions and changes to the timers.

Asleep of two minutes between polls is a lower load on the database and is not too great a sacrifice in terms of synchronizing with the current status of timers and the job queue for the OnceOnly pool.

In a rare case, one would have auction-open timers registered for opening within two minutes of the job registry or have close times extended just two minutes before auction close time. In such rare occurrences, the two-minute sleep could result in unacceptable delayed callbacks. In such cases, decrease the sleep interval to an appropriate value.

```
#CFG_JS_ONCEONLY_SLEEP_INTERVAL  
# - Sleep interval for the OnceOnly timer database thread in  
  milliseconds.  
40=120000 ( Default value = 30000 )
```

Repeat Sleep Interval

The Repeat database polling thread polls the `cmn_job_scheduler_entry` table in order to read any new or revised timers since the last read in order to keep the job queue for the Repeat thread pool in synchronization with additions and changes to the timers.

The two-minute sleep between polls is a good trade off between the load on the database and the synchronizing of the current repeat timer status and the job queue for the Repeat thread pool. The Transactional Mail Scheduler and the RfxJobScheduler repeat timers are registered at module initialization time and are not expected to have any immediate changes after registry.

```
#CFG_JS_REPEAT_SLEEP_INTERVAL  
# - Sleep interval for the Repeat timer database thread.  
41=120000 ( Default value = 60000 )
```

Transactional Mail Scheduler Interval

The Transactional Mail Scheduler (TMS) is a repeatable timer registered by the application for the purpose of sending out email notifications. The time for callback can be configured by a setting in the `VortexConfiguration.properties` file. On callback, the TMS reads the entries in the table `mail_info` and makes calls to the Notification API to send out the emails and deletes the entries in the `mail_info` table after they have been handed over to the Notification API.

The sleep interval for the TMS can be increased from its default value of 30 seconds to 120 seconds. This lowers the database load considerably while sacrificing a bit on the up-to-the-current-second correctness of the email notifications. The time lag between handing over an email notification to the Notifier Thread Pool and its actual delivery depends largely on the settings in the mail transfer agents involved in the delivery chain. Therefore, this sacrifice is negligible if it buys some performance gains.

```
#CFG_TRANSACTIONAL_MAIL_SCHEDULER_INTERVAL
# - Timeout interval parameter for the MailScheduler callback.
  (used for Notifications)
60=120000 ( Default value = 30000 )
```

Max Threads in Notifier Pool

The Notification API hands over the dispatch of all emails to a set of worker threads called the Notifier Thread pool. Increasing the sleep interval of the Transactional Mail Scheduler would result in a larger load of emails the TMS hands over to the Notifier Thread Pool whenever its callback is run. The larger load on the pool necessitates a corresponding increase in the thread pool size. The actual setting will depend on the volume of email notifications expected to be generated in the sleep period for the Transactional Mail Scheduler.

```
#CFG_MAX_THREADS_IN_NOTIFIER_POOL
# - Number of threads in the notifier thread pool.
32=30 ( Default value = 10 )
```

NOTE At various places in this document, it is described how to configure the user interface and the performance parameters by changing the settings in the `VortexConfiguration.properties` and `userxconfig.properties` files. The Installer places two copies of all the properties files in two different locations as follows:

- under `<IMM_HOME>` directory path
- under `<IMM_Install_Dir>` path

Whenever you change the value or the parameter for any property, you should make the changes in both locations and restart the iPlanet WebServer and the iPlanet Application Server.

After the installation of iPlanet Market Maker software, you can find the file `VortexConfiguration.properties` file in the following two locations:

- `<iAS_Install_Dir>/ias/APPS/imm40/imm40/WEB-INF/classes`
- `<IMM_HOME>/resources`

You can find the `userxconfig.properties` file in:

- `<iAS_Install_Dir>/ias/APPS/imm40/imm40/WEB-INF/classes / com/iplanet/userx`
 - `<IMM_HOME>/resources/com/iplanet/userx`
-

Importing and Configuring Data

The iPlanet Market Maker software works with two types of schema and data iPlanet Directory Server (LDAP) and Oracle. Catalogs are stored in the Oracle database.

NOTE The Installer does the configuration, but if you need to import and configure data manually, follow the steps in this section.

This section contains the following sections:

- Creating the LDAP Schema
- Creating the Oracle Schema

Creating the LDAP Schema

1. Source the `mm.kshrc` file.
2. For iDS 4.13:

In the `<iPlanet-Home>/slapd-<instance name>/config/ns-schema.conf` file (where `<iPlanet-Home>` is the directory where you installed the directory server), append these three lines:

```
include <IMM_HOME>/community/ldap/schema/
iplanet-marketmaker-schema.conf

include <IMM_HOME>/community/ldap/schema/
iplanet-userx-schema.conf

include <IMM_HOME>/community/ldap/schema/
iplanet-ecommerce-schema.conf
```

For iDS 5.0 SP1:

Copy

`$IMMHOME/infrastructure/ldap/schema/iDS50SP1/60iplanet-ecs.ldif` to your iDS50SP1 install directory, `<iPlanetHome>/slapd-<instance name>/config/schema`

3. Restart iPlanet Directory Server.
4. For iDS 4.13:

In the directory `<IMM_HOME>/infrastructure/ldap/schema`, there is a file called `createimmsuffix.ldif`, which should be imported before loading the data.

For iDS 5.0 SP1:

In directory `$IMMHOME/infrastructure/ldap/schema/iDS50SP1/` there is a file called `createiMMSuffixDb.ldif`, which should be imported before loading the data

5. Load the LDAP data.

NOTE Presently, there is no utility for bulk loading the community data other than making the data available in an LDIF format conforming to the iPlanet Market Maker LDAP schema.

Deleting the Existing LDAP Data

To delete the LDAP data:

1. Source the environment using the `$IMM_HOME/mm.kshrc` file or the `$IMM_HOME/mm.cshrc` file.
2. If required:

```
cd $IMM_HOME/bin
```
3. Run the `DeleteCommunity.sh` script with hostname `slapd_port`, "cn=Directory Manager" `directory_manager_passwd`.

Creating the Oracle Schema

1. Source the environment using the `$IMM_HOME/mm.kshrc` or the `$IMM_HOME/mm.cshrc` file.
2. Create environment variables as follows:

`VORTEX_SID=<net_service_name>`

`COM_PWD=<schema password>`

`CMN_PWD=<schema password>`

`CAT_PWD=<schema password>`

`AUC_PWD=<schema password>`

`PRI_PWD=<schema password>`

`OMS_PWD=<schema password>`

`EXC_PWD=<schema password>`

`RFX_PWD=<schema password>`

`VORTEX_PWD=<schema password>`

These environment variables will be passed to the execution scripts which will overwrite the parameters in the `installparams.env` file.

3. Run the `$IMM_HOME/setup/bin/installsql.sh` schema script.
4. Run the `$IMM_HOME/setup/bin/installsql.sh` seed script.
5. Import the Oracle sample data:
 - a. Follow the instructions in Steps 1 and 2 above.
 - b. Run the `$IMM_HOME/setup/bin/installsql.sh` sample script.

Running iPlanet Market Maker Software Over SSL (Secure Sockets Layer)

Viewing HTTPS Sites (External URLs) in the iPlanet Market Maker Software

The iPlanet Market Maker software facilitates a user to embed an external site on any iPlanet Market Maker page.

HTTP sites are non-secure and can be embedded on any iPlanet Market Maker page, while HTTPS sites are secure and cannot be embedded.

The URL class in JDK1.2.2 does not recognize HTTPS as a legal protocol. When using JSSE 1.0.2, the HTTPS protocol becomes a legitimate protocol for JDK1.2.2. It can then be recognized to access any secure site, as long as the certificate is installed on the server.

Install and Customize the iPlanet Market Maker Software for Secure Site Access

NOTE Installation and Customization should be done on the machine where the iPlanet Market Maker software is installed.

1. “Download JSSE 1.0.2 from the Sun Site”
2. “Install JSSE Pack”
3. “Install the Certificate of the Secure Site”
4. “Set the Security Provider”
5. “Set the Configuration Parameters”

Download JSSE 1.0.2 from the Sun Site

The JSSE 1.0.2 pack can be downloaded from the following site:

<http://java.sun.com/products/jsse/index-102.html>

JSSE 1.0.2 is supplied as an extension to the java-2 platform. JSSE is implemented via a Java Cryptography Architecture (JSA) security provider class called `sunJSSE`.

NOTE Choose the appropriate path separator for Windows and Solaris.

When `<ias-java-home>` is used in this document, it refers to the `\iplanet\ias6\ias\usr\java\jre` directory.

Install JSSE Pack

JSSE is a zip file. When the zip file is extracted, it creates the `jsse1.0.2` directory, with two subdirectories named `doc` and `lib`.

1. Select the `jsse.jar`, `jcrt.jar`, and `jnet.jar` files from the `jsse1.0.2\lib` directory.
2. Copy them to the `<ias-java-home>\lib\ext` directory.

Install the Certificate of the Secure Site

Install a JSSE-specific cacerts file.

It can be installed in the `<ias-java-home>\lib\security` directory with the name `jssecacerts`.

Sun's JSSE implementation checks for an alternate cacert file before falling back on the standard cacerts file, so you can provide a JSSE-specific set of trusted root certificates separate from the ones that might be present in cacerts for code signing purposes. The search order for the locating the trust store is:

1. `<ias-java-home>/lib/security/jssecacerts`
2. `<ias-java-home>/lib/security/cacerts`

If the file `jssecacerts` exists, then `cacerts` is not consulted.

Set the Security Provider

JSSE 1.0.2 comes standard with a Cryptographic Service Provider, or *provider*, named *SunJSSE*. Although the *SunJSSE* provider is supplied with every JSSE 1.0.2 installation, it still must be configured explicitly before its services can be accessed.

Add the SunJSSE Provider to Your List of Approved Providers

To add the security provider statically, add the following statement in `<ias-java-home>/lib/java.security` file:

```
security.provider.2=com.sun.net.ssl.internal.ssl.Provider
```

NOTE Do not delete
 `security.provider.1=sun.security.provider.Sun` that is the
 default security provider existing in the `java.security` file.

Set the Configuration Parameters

Open the `vortexconfiguration.properties` file and set the following properties:

```
java.protocol.handler.pkgs=com.sun.net.ssl.internal.www.protocol
```

If you use proxy:

```
https.proxyHost=<your proxy host>
```

```
https.proxyPort=<your proxy port number>
```

These properties are read by the

`com.iplanet.ecommerce.vortex.arch.JvmSettings` class and set as the system properties.

Performance Tuning and Monitoring

After installing the iPlanet Market Maker software, a number of performance tuning steps can be carried out to ensure the best performance of the application.

This chapter contains the following sections:

- Testing Scenarios
- Tuning the System
- Tuning iPlanet Directory Server
- Tuning Oracle
- Tuning the iPlanet Application Server
- Tuning iPlanet Market Maker
- Monitoring Performance

Testing Scenarios

The performance and tuning recommendations in this guide are based on findings from tests conducted in our test labs to measure the performance of the iPlanet Market Maker software in different scenarios. Performance tests were conducted on individual modules and on mixed scenarios. Mixed scenarios simulated the realistic conditions of the initial phase of deployment as well as operation in a production environment. The tests focused on the most commonly used functions in each module.

The performance tests were conducted on different sizes of catalog and community. Some examples of test cases:

- Catalog search

- Catalog browse
- Community - Registration

Tuning the System

The following sections contain information on tuning your system to improve performance:

- Tuning Solaris Kernel Parameters
- Configuring Memory
- Configuring Disks

Tuning Solaris Kernel Parameters

To achieve better performance and scalability, the Solaris operating system can be tuned. These settings should be made on the client iPlanet WebServer/iPlanet Market Maker hosts. The following settings should be made in the system file so that the settings become effective with the current values whenever the server is booted:

```
Set rlim_fd_max=8192
Set rlim_fd_cur=8192
Set sq_max_size=0
```

Run the following settings from the command line whenever the system is rebooted (they can be grouped and run as a script file):

```
ndd -set /dev/tcp tcp_time_wait_interval 60000
ndd -set /dev/tcp tcp_conn_req_max_q 1024 (Only for Solaris 2.8)
ndd -set /dev/tcp tcp_conn_req_max_q0 4096
ndd -set /dev/tcp tcp_ip_abort_interval 60000
ndd -set /dev/tcp tcp_keepalive_interval 900000
ndd -set /dev/tcp tcp_rexmit_interval_initial 3000
ndd -set /dev/tcp tcp_rexmit_interval_max 10000
ndd -set /dev/tcp tcp_rexmit_interval_min 3000
ndd -set /dev/tcp tcp_smallest_anon_port 1024
```

```
ndd -set /dev/tcp tcp_slow_start_initial 2
ndd -set /dev/tcp tcp_xmit_hiwat 32768
ndd -set /dev/tcp tcp_recv_hiwat 32768
```

Configuring Memory

Sizing the RAM for your configuration depends on the implementation scenario you choose. The only reliable way to estimate your memory requirements is through stress testing the system with the user loads and user actions you would expect in a real world scenario.

Each server type in the production systems have different forces that drive the memory requirements. Consider the following factors to configure memory across different servers and perform stress test to monitor the memory utilization.

- The proxy tier utilizes memory for caching and SSL decryption. The decryption proxy servers utilize memory for decrypting the content sent to and returning from the web server layer. The size of the memory which can only be determined by stress and performance testing should be tailored to the throughput of the proxy server layer.
- The static server is responsible for caching the static content of the market place. The more static content you have that can be cached, the more you will benefit from more memory.
- The application server tier derives its memory requirements more directly from the application itself. This is determined by the marketmaker in addition to the customization that is made to the product in your installation. As you increase the number of users on the system, the memory requirements for the application server increase. As a general rule, most application servers are configured for 512MB to 1 GB of RAM per CPU. Your selection of the appropriate amount of RAM will be contingent on your performance testing results.
- The iPlanet Directory Server benefits a great deal from memory and so it is important that the memory be sized appropriately for your installation. The memory in the iPlanet Directory Server used for caching is able to deliver very high levels of performance when ample memory is available. Refer to the iPlanet Directory Server documentation for latest sizing instructions.

- The Oracle 8.1.7 / 9i server also benefits greatly from appropriate amounts of memory. Since the sizing of the Oracle 8.1.7 / 9i server can be very complex, consult your DBA for information on sizing the memory on the server.

Refer to “Monitoring Performance,” on page 90 to learn how to observe memory utilization. A full listing on all the processes with the processor numbers and their memory utilizations can be helpful in determining the memory configuration. A few tips:

- Running the `vmstat` command will give you the information about the free and available memory that can be utilized optimally.
- At the application server layer, if you are running short of memory, you can down size JVM heap, taking care not to go below 256MB.
- On the database machine, give Oracle as much memory as you can to limit or avoid swapping.

For more details on configuring the memory, refer to the administrator’s guides of the individual server applications.

Configuring Disks

Depending on the size of the database, read/write loads can be heavy. Placing the Oracle tablespaces and LDAP database on a single disk will result in queueing the read and write operations, causing undue delay in response. Separating and distributing the Oracle tablespaces among multiple disks or an array of disks enables simultaneous (parallel) access to the database, which will solve the problem. The iPlanet Market Maker software is configured to work with separate Oracle data and index tablespaces for each module, such that the tablespaces can be distributed among separate disks or disk arrays. For example, the Auction, and Catalog modules have separate data and index tablespaces. While bidding on auctions, a user accesses the Auction tables, the Catalog tables, and the Common tables individually. If the tablespaces for the data and their indexes are placed in separate disks or disk arrays, they can be accessed simultaneously, which improves response time considerably. Similarly, LDAP database files, index files and transaction logs can also be distributed among multiple disks to achieve better response times during Community-Registration activity.

Based on our performance analysis, the following general guidelines are recommended:

- **Disk array striping**—Striping the group of disks creates a logical mount point. These striped logical disk units can be used in place of singular disks for better performance. Hardware and software striping can be used to get optimal performance by striping the groups of disks. The disk arrays can be mirrored by using the operating system’s mirroring or RAID solutions.
- **I/O distribution**—Distribute the I/O evenly among the controllers.

For specific instructions on how to distribute Oracle and LDAP database among multiple disks, refer to the following sections:

- “Distributing the Write Load Among Multiple Disks,” on page 76
- “Distributing the Tablespace Among Multiple-Disks,” on page 82

Tuning iPlanet Directory Server

To improve response times, read and write performance of the LDAP and Oracle databases must be configured. The following sections describe how LDAP and Oracle can be configured to achieve better performance. Also refer to iPlanet Directory Server documentation and “Oracle 8.1.7 / 9i Designing and Tuning for Performance”.

Improving the LDAP Read Performance

LDAP maintains two levels of cache. Improved LDAP performance can be achieved by tuning these two-level caches:

- Database cache
- Entry cache

The basic rule for sizing these caches is:

Dedicate 75% of your cache memory to `dbcachesize` and 25% to the entry cache size. For example, if you have 500 MB RAM available for caching the directory, tune the caches as described below.

Tuning the Database Cache

The database cache is the lower-level cache, which caches the pages from the database. The size of the database cache is set in terms of memory size in the file `slapd.ldbm.conf` in the config directory of the LDAP instance home.

1. Calculate the database cache as follows:

The memory used by the database cache = $500 * 75\% = 375\text{MB}$, and the overhead associated with cache management = 25% ; therefore the setting should be = $375\text{MB} / 1.25 = 300 \text{ MB}$

2. Apply the caveats as follows before accepting the results:
 - o If the result is larger than 1.6 GB, reduce it to 1.6 GB (the maximum dbcache size).
 - o If the result is smaller than the total size of the database indexes (calculated by adding the sizes of all *.db2 files in the database directory and deducting the size of the file, id2entry.db2), increase it to match the size of the indexes, taking care not to exceed available memory.
3. Set the size in this file:

```
<iPlanet-home>\slapd-<instance name>\config\ slapd.ldbm.conf
```

Tuning the Entry Cache

The entry cache is the higher-level cache, which caches the most recently used entries from the directory. The size is set in the `slapd.ldbm.conf` file in the config directory of the LDAP instance home, in terms of the number of entries it should hold rather than a memory size.

1. Calculate the entry cache as follows:

Required Entry size = 1000 bytes per entry.

Memory used by entry cache = $500 * 25\% = 125\text{MB}$;

Available memory including overhead = $125 / 1.25 = 100\text{MB}$;

Therefore, the setting should be = $100,000,000 \text{ bytes} / 1,000 \text{ bytes per entry} = 100,000 \text{ entries}$.

2. Set the size in this file:

```
<iPlanet-home>\slapd-<instance name>\config\ slapd.ldbm.conf
```

Improving the LDAP Write Performance

LDAP write performance can be improved by the methods discussed in the following sections:

- Setting the `db_home_directory`

- Distributing the Write Load Among Multiple Disks
- Disabling Durable Transactions
- Setting the ALL_IDS_THRESHOLD

Setting the db_home_directory

Making this setting reduces the LDAP-write related response times considerably. The standard deviations in the response times would decrease considerably. Though the average response times are not considerably affected, this setting would reduce isolated peaks observed in the LDAP write timers thus reducing the chances of a few users being timed out during the registration process under load.

This method can be used if the following symptoms are observed from vmstat and iostat on the LDAP host:

- The operating system seems to flush pages endlessly. This flushing can be so excessive that performance of the entire system is severely degraded.
- The disk is heavily used (more than 1MB per second of data transfer).
- There is a long service time (more than 100ms).

Two additional conditions where this setting may be necessitated are:

- The database cache size is around 100 MB or more.
- There is mostly write activity going on the LDAP host.

To set the db_home_directory:

The directory referenced by db_home_directory must be a subdirectory of a file system of type tempfs (such as /tmp).

1. Create a subdirectory manually under /tmp.
2. Set db_home_directory to /tmp/<subdirectory> in the file:

```
<iPlanet-Home>\slapd-<instance name>\config\ slapd.ldbm.conf
```

The above setting causes the internal directory server database files to be moved to the directory referenced by the parameter.

Caution: It is possible, but unlikely, that the server will no longer start after the files have been moved, because enough memory cannot be committed. This is a symptom of an overly large database cache size being configured for the database server. If this happens, reduce the size of the database cache size to a value at which the server will start again.

Making this setting will considerably smoothen the LDAP-write related response times. Though there will not be any considerable change in the average response times, it would reduce isolated peaks in the LDAP-write timers, thus reducing the chances of a few users being timed out.

Distributing the Write Load Among Multiple Disks

Heavy LDAP write loads can occur if there is heavy registration activity. Placing the LDAP database on a single disk can cause long write queues (serial access). Under heavy write loads, this can result in long response times. A multiple-disk system facilitates fast and simultaneous (parallel) access to the database and will reduce the write queues. In “Configuring Disks,” on page 72, you will find the guidelines to configure multiple disks or disk arrays. After configuring the multiple-disk system, you can distribute the LDAP database files, index files, and transaction logs across disks to achieve better response times for the registration activity.

The default installation of the directory places all directory data files under a single subdirectory. These files include:

- Database and index files
- Transaction logs used mainly for recovery purposes
- LDAP access, error, and audit logs

To store the database and log files on different disk volumes, set the directives in the data files `slapd.conf` and `slapd.ldbm.conf` to different disks.

1. Modify the directory setting in the

```
<iPlanet-Home>/slapd-<serverID>/config/slapd.ldbm.conf file to point  
to a different location from its default setting of  
<iPlanet-Home>/slapd-<serverID>/db.
```

2. The default setting for the transaction log files is:

```
<iPlanet-Home>/slapd-<serverID>/db
```

Modify the `db_logdirectory` setting in the `<iPlanet-Home>/slapd-<serverID>/config/slapd.conf` file to point to a different location on a different disk from the one on which the database and index files are located.

3. In the `<iPlanet-Home>/slapd-<serverID>/config/slapd.conf` file, modify the `accesslog` setting to point to a different location on a file system mounted on a disk different from that on which the database, index, and transaction log files are set up.
4. When the installation achieves some stability, consider turning off LDAP access logs, because LDAP is very heavy in the writes to the access logs.

In the `<iPlanet-Home>/slapd-<serverID>/config/slapd.conf` file, turn off access logging by setting the `accesslog-logging-enabled` parameter to `Off`.

Disabling Durable Transactions

To reduce the response time for the registration activity and also to reduce the write intensiveness on the disk to which the LDAP transaction logs are mapped, set `db_durable_transactions` to `Off` in the `<iPlanet-Home>/slapd-<serverID>/config/slapd.ldbm.conf` file. This setting is expected to provide some performance gains in terms of better response times under conditions of high load of the registration activity on the site.

Quoting the *Administrator's Guide* for iPlanet Directory Server:

“By default, durable database transaction logging is enabled. This means that every time a write is performed on the directory, a corresponding entry is physically written to the database transaction log disk. To improve performance, you can disable durable transaction logging. When you do so, every directory database operation is logically written to the database transaction log file, but it may not be physically written to disk immediately. That means that if a directory change was written to the logical database transaction log file but not physically written to disk at the time of a system crash, you cannot recover the change. When durable transactions are disabled, the recovered database is consistent, but does not reflect the results of any LDAP write operations that completed just before the system crash.”

Setting the ALL_IDS_THRESHOLD

If the community size (number of companies) exceeds the value set for ALL_IDS_THRESHOLD, a search with a filter such as ou=immOrg turns into an unindexed search, causing the search times on some community pages to be inordinately long. The default value for this setting is 4000. If the community size grows beyond 4000, considerable deterioration in performance can be expected in terms of response time for login.

The solution to this is to set the ALL_IDS_THRESHOLD to a value equal to the community size that is expected to be reached at steady state for the site.

Refer to the iPlanet Directory Server *Administrator's Guide* for more details on setting the appropriate value for ALL_IDS_THRESHOLD.

1. Set the ALL_IDS_THRESHOLD to the required value in the `<iPlanet-Home>/slapd-<serverID>/config/slapd.ldbm.conf` file.
2. Refer to the following for more information about tuning the LDAP and make the appropriate changes in settings to improve write performance on LDAP:
 - o Administrators reference at:
<http://docs.iplanet.com/docs/manuals/directory/41/admin/>
 - o LDAP Performance Tuning Primer at:
<http://mc.red.iplanet.com/launches/directory/dir4launch/performance/PerfTuningPrimer.html>

LDAP Indexes

While importing LDAP schema and data during the installation of iPlanet Market Maker software, LDAP indexes are created. The `slapd.ldbm.conf` file holds the list of indexes that are created on the LDAP database. While logging in or while performing any such task involving the LDAP database, indexed searches will be made if the indexes are present. Absence of any index will make it an unindexed search, which are very slow. The Login page and also some other pages where calls to LDAP are made for authenticating/selecting available privileges for a user will become much slower if the search is unindexed.

If the indexes on any of the following attributes are missing, notes=U entries will appear in the LDAP access logs corresponding to LDAP searches where the search filter includes any of the unindexed attributes.

The indexes listed in Table 4-1 are created. You can verify whether they are present in the `slapd.ldbm.conf` file.

Table 4-1 LDAP Indexes

Index Name
index aci pres
index changenumber eq
index cn pres,eq,sub
index copiedfrom pres
index displayName sub
index dncomp eq
index entrydn eq
index givenName pres,eq,sub
index immBusinessRole eq
index immMembershipStatus eq
index immroletype eq
index iplanetecguid eq
index iplanetecparentorgguid eq
index mail pres,eq,sub
index mailAlternateAddress eq
index mailHost eq
index member eq
index memberof eq
index nsActionName eq
index nsAllowedPrincipal eq
index nsCalXItemId pres,eq,sub
index nsLIProfileName eq
index nsResourceName eq
index nswcalCALID pres,eq
index ntGroupDomainId pres,eq,sub
index ntUserDomainId pres,eq,sub
index numsubordinates pres

Table 4-1 LDAP Indexes (*Continued*)

Index Name
index o eq,sub
index objectclass eq
index ou eq
index owner eq
index parentid eq
index pipstatus eq
index pipuid pres,eq,sub
index seeAlso eq
index sn pres,eq,sub
index telephoneNumber pres,eq,sub
index uid pres,eq,sub
index uniquemember eq
pres—presence; eq—equality and sub—substring index

To add an index on an attribute in LDAP when LDAP already has data loaded, you can either use the Directory Server Console or make suitable manual additions to the `slapd.ldbm.conf` file.

Adding indexes through the Directory Server Console is the suggested method because addition of indexes through the console ensures that the result takes effect on the existing LDAP database retrospectively. Note that while adding indexes from the console, the server is placed in a read only mode. During this process, we can neither make any configuration changes to the server nor modify the contents of the directory.

The other method of making manual edits to the file `slapd.ldbm.conf` file is as follows:

1. Add these attributes manually to `slapd.ldbm.conf` file.
2. Create indexes on these attributes for the existing data in LDAP if it has any entries that are unindexed with respect to these attributes. This can be done by one of the two methods given below:

- o Export the database as LDIF and import it again. This will cause the indexes to be updated and applied on all the new additions in the file `slapd.ldbm.conf`.

OR

- o Run the `db2index` command line tool.
- a. Stop the server, or place LDAP in read-only mode as follows:
 - I. On the Directory Server Console, select the Configuration tab.
 - II. Select the Database icon in the navigation tree in the left panel.
 - III. Select the Settings tab in the right panel.
 - IV. Select the Make Database Read-Only check box.
 - V. Click Save.
- b. Change the directory to `<iPlanet-Home>/bin/slapd/server`, where `iPlanet-Home` is the directory where you installed the directory server.

In this directory, you can find a utility called `ns-slapd`.

- c. Run the `ns-slapd` process as follows:

```
ns-slapd db2index -f <slapd.conf> -t
attributeName[:indexTypes]
```

Where:

- `attributeName` is the name of the attribute to be indexed
 - `indexTypes` is a comma-separated list of indexes to be created for the attribute
 - `slapd.conf` is the full path to the `slapd.conf` file
- d. Restart iPlanet Directory Server.
 - If iPlanet Directory Server was running at the time of index creation, stop the server, take it out of read-only mode, and restart it.
 - If it was stopped for index creation, simply restart it.

VLV Indexing

The Virtual List View (VLV) is used to implement pagination. The iPlanet Market Maker application uses VLV control to request subsets of search results, so they can be sorted and displayed by the pages. VLV indexing allows efficient browsing of millions of entries on the search filters that are indexed. To allow efficient browsing of millions of entries, you should create a special VLV index for the query.

The VLV indexes are available only for searching and sorting (ascending/descending) the companies on the status filters Pending, Active, and All.

The VLV index is created initially by the Installer after schema creation. A VLV index creation script (`vlvindexes.bash`) is placed in the `$IMM_HOME/infrastructure/ldap/schema` directory. For more information on VLV indexes and its usage in iPlanet Market Maker.

For more information, refer to `vlvindex.README` in the `$IMM_HOME/infrastructure/ldap/schema` directory.

Tuning Oracle

Distributing the Tablespaces Among Multiple-Disks

The iPlanet Market Maker software is configured to work with separate Oracle data and index tablespaces for each module, such that the tablespaces can be distributed among separate disks or disk arrays. In “Configuring Disks,” on page 72, you can find instructions for configuring a multiple-disk subsystem. After configuring multiple disks or disk arrays, the data and index tablespaces can be distributed across multiple disks or disk stripes. Some recommendations based on the tests carried out in our performance labs:

- Place data tablespaces and index tablespaces on separate disks or a disk stripe.
- Separate heavily used tablespaces, such as catalog, auction, and common (cmn), so that the tables can be accessed simultaneously.
- Place redo logs on a disk separate from other disks.
- Create rollback segments based on the concurrent transactions and separate rollback tablespaces.
- Place temporary tablespaces on disks separate from other tablespaces.

- Place system tablespaces on disks separate from other tablespaces.

After the installation, the Oracle data files can be moved to different locations to balance I/O.

Assume that an open database has a tablespace named USERS that is made up of data files located on the same disk of a computer. To relocate the data files of the USERS tablespace to a different disk drive, run the following steps:

1. Find the data files that need to be relocated.

The following query of the data dictionary view DBA_DATA_FILES lists the data file names and sizes (in bytes) of the USERS tablespace:

```
SELECT file_name, bytes FROM sys.dba_data_files WHERE
tablespace_name = 'USERS';
```

This will return:

```
FILE_NAME BYTES
-----
/U02/ORACLE/RBDB1/USERS01.DBF 102400000
/U02/ORACLE/RBDB1/USERS02.DBF 102400000
```

2. Back up the database. Before making any structural change to a database, such as renaming and relocating the data files, it is always recommended that the entire database is backed up.
3. Take the tablespace containing the data files offline, or shut down the database and restart and mount it, leaving it closed. Both the options close the data files.
4. Copy the data files to their new locations and rename them using operating system commands. You can also run an operating system command to copy a file by using the `HOST` command.
5. Rename the data files within Oracle.

The data file pointers for the files that make up the USERS tablespace, recorded in the control file of the associated database must be changed from the old names to the new names.

- a. If the tablespace is offline but the database is open, use the `ALTER TABLESPACE...RENAME DATAFILE` statement.

- b. If the database is mounted but closed, use the ALTER DATABASE...RENAME FILE statement.

```
ALTER TABLESPACE users
        RENAME DATAFILE '/u02/oracle/rbdb1/users01.dbf',
        '/u02/oracle/rbdb1/users02.dbf'
        TO '/u03/oracle/rbdb1/users01.dbf',
        '/u04/oracle/rbdb1/users02.dbf';
```

- 6. Bring the tablespace online, or shut down and restart the database.
 - a. If the USERS tablespace is offline and the database is open, bring the tablespace back online.
 - b. If the database is mounted but closed, open the database.
- 7. Back up the database. After making any structural changes to a database, always perform an immediate and complete backup.

Refer to the Oracle 8.1.7 / 9i *Administrator's Guide*.

Tuning the iPlanet Application Server

The installations of the iPlanet Application Server should be tuned to achieve optimal performance. This section describes the settings specific to the iPlanet Market Maker application.

- 1. Apply the JDK1.3.3 patch provided by JavaSoft for performance improvement with the `-server` option. Otherwise, response times will be unsatisfactory.
- 2. Set `JAVA_ARGS` in `IAS_HOME/ias/env/iasenv.ksh` to the following:

```
JAVA_ARGS="-server -Xms256m -Xmx608m -XX:NewSize=96m
-XX:MaxNewSize=96m -XX:SurvivorRatio=2
-XX:+UseLWPSynchronization"
```

- 3. Make a file in `IAS_HOME/ias` as `.hotspot_compiler`.

Add the following two entries in the `.hotspot_compiler` file:

```
exclude oracle/sql/NUMBER_isPositive
exclude com/iplanet/ecommerce/vortex/community/UxUtilImpl
populateDAG
```

```
exclude
com/iplanet/ecommerce/vortex/arch/PersistentLocalizedMessage
create
```

4. Do not change the values for the minimum and maximum threads for the kjs processes thru the admin tool.

Use the defaults: Minimum threads=8 and Maximum threads = 64.

Also leave the Request Queue Low Water Mark and Request Queue High Water Mark to default values.

Creating New KJS Processes and Tuning

On a multi CPU machine, you can set side one CPU for every 4 CPUs for processes other than KJS such as `ns-httpd`, `ns-slaped` and other system monitoring tools. So, it is recommended that for every 4 CPUs, you should create three KJS processes.

1. Log in as the user for the catalog module.
2. Change directory to:


```
<iAS_HOME>/ias.0sp2/ias/bin
```
3. Run `ksvradmin`.
4. Select the General tab in the iPlanet Application Server Administration Tool.

5. Select the iPlanet Application Server instance and create a new KJS process as shown in the picture below.



Session Model

We recommend the following iPlanet Application Server session data storage model for the optimal performance of iPlanet Market Maker.

```
<impl>=lite <dsync-type>=ignored <sticky>=true
```

Source of Session Data = KJS

Advantages

- Fast access—Session data is sourced in process with request.
- Fast access, and no restrictions on types of objects that can be store in the session—No serialization/deserialization.

Disadvantages

- No load balancing—Session data is only accessible from original KJS.
- No failover—Session data is lost if KJS crashes.

- No backward compatibility—J2EE components only; AppLogics are not supported.

The iPlanet Market Maker Installer places two XML descriptor files, `ias-web.xml` and `web.xml`, under `APPS\imm40\imm40\WEB-INF`.

```
<session-info>
<impl>lite</impl>
</session-info>
```

The Installer sets the `<impl>lite</impl>` and `<sticky>>true</sticky>` in the `ias-web.xml` file, and sets the `<distributable>>false</distributable>` in the `web.xml` file.

To ensure that it is registered in the registry, you should launch the `kregedit` and change the default value for `distributable` to `false` as follows:

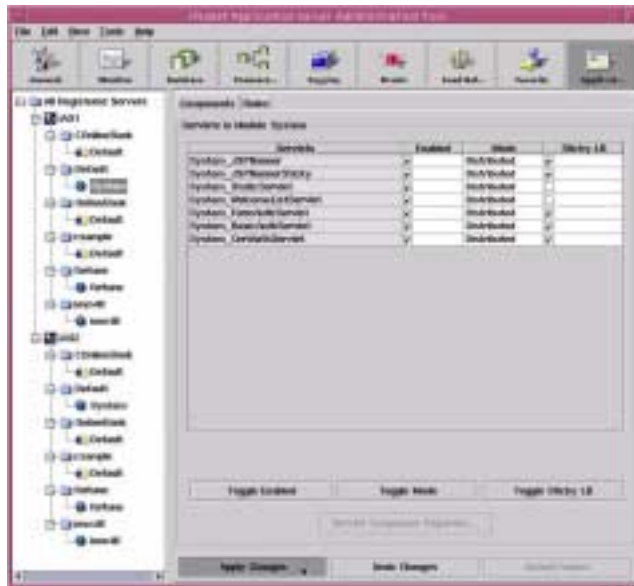
```
SOFTWARE/iPlanet/Application Server/6.0/J2EE-Module/imm4.0/
distributable=false
```

These settings along with the iPlanet Market Maker classes are registered in iPlanet Application Server registry data file, `reg.dat`, by invoking the `iasdeploy` command.

Load Balancing

Load balancing is session based. Session data is stored in the KJS. The first request in a session is load balanced. All subsequent requests in that session are sent to the server chosen for the first session request. To enable sticky load balancing for JSPs, including `imm.jsp`, complete the following steps. Click the Application icon in the Administration Tool to view and configure application settings.

1. Select the iPlanet Application Server instance where iPlanet Market Maker is installed as an application; navigate to the system module under Default application.
2. On the right pane, the servlets are displayed. Ensure that the Sticky Load Balance check boxes for servlets `System_JSPRunner`, and `System_JSPRunnerSticky` are checked.
3. Similarly, select the module `imm40` under application `imm40` and ensure that the Sticky Load Balance checkbox is checked.
4. Click Apply Changes.
5. Restart the iPlanet WebServer.



To Use round robin load balancing:

1. Select the tab Load Bal.... in the iPlanet Application Server Administration Tool.
2. Select an iPlanet Application Server instance and select the Round Robin (connector-driven) option for load balancing from the drop-down menu.
3. Repeat the above step for each instance of iPlanet Application Server.
4. Click Apply Changes.
5. Restart the Application Server.

Alternatively, you can perform the above settings by editing the registry as follows:

6. Run `kregedit`.
7. Add/modify the values for load balancing as follows:

`SOFTWARE/iPlanet/Application Server/6.0/CCS0/LOADB/Disable=0`
(Integer)

`SOFTWARE/iPlanet/Application
Server/6.0/CCS0/LOADB/RoundRobin=1`(Integer)


```
SOFTWARE/iPlanet/Application
Server/6.0/CCS0/LOADB/ServerWeights/iAS1.domain.com(or IP Address
1)=1(Integer)
```

```
SOFTWARE/iPlanet/Application
Server/6.0/CCS0/LOADB/ServerWeights/iAS2.domain.com(or IP Address
2)=1(Integer)
```

```
SOFTWARE/iPlanet/Application
Server/6.0/CCS0/LOADB/ServerWeights/iAS3.domain.com(or IP Address
3)=1(Integer)
```

8. Restart the iPlanet Application Server.

NOTE To apply changes, whenever you restart iPlanet Application Server, use the `iascontrol kill` and `iascontrol start` commands instead of the `iascontrol stop` and `iascontrol start` commands.

Setting the Heap Size for JVM

It is recommended that you set the minimum and maximum Java heap size in the KJS script as 256MB and 608MB respectively. Modify the KJS script for the JAVA heap size and garbage collection parameters as follows:

```
JAVA_ARGS="-server -Xms256m -Xmx608m -XX:NewSize=96m
-XX:MaxNewSize=96m -XX:SurvivorRatio=2
-XX:+UseLWPSynchronization"
```

It is not recommended to set this value lower than 256MB. You can set this to as high a value as necessary depending on the available RAM.

Tuning iPlanet Market Maker

Turning Off the Pricing Engine

For multiple users, the pricing engine has a significant impact on performance. If you are not actively using the pricing engine, it can safely be turned off by editing the following file:

```
<appserver_dir>/ias/APPS/imm40/imm40/WEB-INF/classes/CatalogConfiguration.properties
```

1. Set the `CFG_NO_ADJUSTED_PRICES` flag to true (8=true).
2. Restart the iPlanet Application Server.

Monitoring Performance

Monitoring performance is helpful in sizing and performance tuning. A well-tuned system should indicate:

- An even usage of CPU time across all of the servers
- An even usage of CPU time across all processors in each server
- A fairly low percentage of system time (0 - 25%), especially if the workflow is intensively computational

If you find requests waiting while the processors are not 100% busy, it indicates that the server is not tuned optimally. Check the active database connections to find the appropriate number of connections for the database connection pooling.

Monitoring Tools

A number of command-line tools are available to monitor various performance parameters on Solaris.

Monitoring I/O Usage

Run the following command to get I/O statistics:

```
/usr/bin/iostat
```

Monitoring CPU Utilization

Run the following command to get virtual memory statistics regarding process, virtual memory, disk, trap, and CPU activity:

```
vmstat
```

Monitoring Process Resource Utilization

Run the following command to get process statistics:

```
prstat
```

NOTE Available only on Solaris 8. You can use Top as an alternate to prstat.

Monitoring Multi-Processor CPU Usage

Run the following command to get statistics on each processor:

```
/usr/bin/mpstat
```

Observing Memory Usage

Run the following command to get a report on process status:

```
# ps -Pefly | grep "\.k"
```

This gives a long full listing about every process with the processor number and memory utilization.

Monitoring Network Activity

To watch network activity between IP nodes (server1 and server2), run the following command:

```
# snoop from <server1> to <server2>
```

Log Files

Logging can be very useful in troubleshooting and solving problems. The web server logs will give you a good idea of activities taking place at a lower level, (for example, JVM issues). The iPlanet Market Maker logs will give you a great deal of application-specific messages, which will be invaluable as you debug a problem. Fresh logs are created every time the iPlanet WebServer JVM restarts.

While debugging a problem, it is a good idea to `tail` the output of the iPlanet Market Maker log file to watch for errors as JSP pages are returned.

You can adjust the debug settings for the iPlanet Market Maker installation to get a more detailed internal view of the application.

iPlanet Market Maker Logs

The iPlanet Market Maker logs are located under the `imm/logs` directory. You can set the debug level in the file `VortexConfiguration.properties` file.

```
#CFG_DEBUG_LEVEL (off=3, terse=7, verbose=15, vverbose=31)
# - Debug level for the Market Maker system.
5=3
```

The default setting of OFF means that only errors will be logged. It is recommended not to increase the debug level to any point above `terse`, because the log volume will quickly be overwhelming. Be sure to set debug to OFF when running under a load, as it will have a dramatic negative impact on performance if debug remains ON.

Every time you recycle the iPlanet WebServer, it creates a new log file with a time stamp. The location of the debug log files is specified in the file `VortexConfiguration.properties` as follows:

```
#CFG_DEBUG_DESTINATION
# - Location of the debug log file.
6=@IMM_HOME@/logs/imm.log
```

iPlanet Application Server Logs

Log files can help you monitor server activity and troubleshoot problems.

To configure logging options for iPlanet Application Server:

1. Access the administration tool and choose the Logging tab.

Here, you can choose to log Errors only, Errors and Warnings, or All Messages. You can choose to log the messages to a database, console, or a file. For better performance, it is recommended that you disable logging.

2. Deselect the Enable Server Event Log option to ensure best performance. With this setting, the exceptions from applications will still be reported in KJS logs.

To view the error log file:

- To view application logs, open KJS logs from the `<iAS-HOME>/logs` directory, and open iPlanet Market Maker logs from the `<IMM-HOME>/logs` directory.
- To troubleshoot iPlanet Application Server problems, open `kxs.log`, `kas.log` and `ias.log` files from the `<iAS-HOME>/logs` directory.

For more details on logging options, refer to the iPlanet Application Server Administrator's Guide.

iPlanet Directory Server Logs (LDAP)

Access, audit, and error logs are generated by LDAP. A number of settings related to logging can be made in the `<iPlanet-Home>/slapd-<server ID>/config/slapd.conf` file. When the installation achieves some stability, turn off access logging by setting `accesslog-logging-enabled` to `Off`.

Verifying the Deployment

This chapter discusses how to verify that your deployment of the iPlanet Market Maker software has been successful. Guidelines for ongoing DBA maintenance activities are also included.

The following sections are contained in this chapter:

- Deployment Verification Checklist
- Recommended DBA Maintenance Activities

Deployment Verification Checklist

Table 5-1 lists the tasks you need to perform to verify that your system has been installed and configured properly.

Table 5-1 Verification Checklist

Component		Description
Solaris	1	Verify that the correct patches for Solaris are installed on all the servers as specified in the iPlanet Market Maker <i>Installation Guide</i> .

Table 5-1 Verification Checklist (*Continued*)

Component		Description
Oracle	2	Verify that the initial database administration setup is performed and that the initialization parameters are set to values corresponding to your catalog size.
	3	Perform periodic maintenance—Apart from daily database monitoring and logging activities, analyze the tables and indexes and gather statistics. Run the <code>ctxsrv</code> process.
	4	Verify that Net8 is configured for the Installer to create the Oracle schema. Verify that SQLPLUS can connect as follows: <code>sqlplus userid/passwd@service_name</code>
	5	Verify that the database URL points to the correct host and port locations on all the iPlanet WebServer/iPlanet Market Maker hosts.
	6	Verify that the maximum connection pool sizes are set to correct values on all the iPlanet WebServer/iPlanet Market Maker hosts.
	7	Verify that the extent sizes of the Oracle tables (indexes) for different modules are set to appropriate values.
iPlanet Directory Server (LDAP)	8	Verify that <code>ldaphost</code> and <code>ldappport</code> settings point to the correct host and port locations on all the iPlanet WebServer/iPlanet Market Maker hosts.
	9	Verify that the minimum and maximum values of the LDAP connection pool settings are set correctly on all the iPlanet WebServer/iPlanet Market Maker hosts.
	10	Verify that the LDAP indexes are added to the <code>slapd.ldbm.conf</code> file.
	11	After the system achieves stability, verify that the LDAP access logs are turned off.

Table 5-1 Verification Checklist (*Continued*)

Component		Description
iPlanet Application Server	12	Verify that the maximum heap size and the garbage collection parameters for JVM in the KJS script are set to prescribed values on all iPlanet Application Server instances.
	13	Verify that the maximum and the minimum of threads for KJS and kxs is set to 32.
	14	For optimal performance, verify that Enable Server Eventlog is unchecked in the Logging tab of the administration tool.
	15	Verify that the Sticky Load Balance for servlets System_JSPRunner and System_JSPRunnerSticky are enabled.
	16	Use the <code>kregedit</code> command to verify that the round robin parameters under the LOADB for each instance of iPlanet Application Server is set to 1.
	17	For optimal performance, use the <code>kregedit</code> command to verify that the distributable session parameter is set to false.
iPlanet Market Maker	18	Verify that job scheduler is enabled.
Proxy servers	19	Verify that the forwarding rules are set so all requests are passed from the proxy server to the web server.
Firewalls	20	Verify that all ports other than port 80 are disabled.
Interoperability	21	If iPlanet Market Maker and iPlanet BuyerXpert are running on separate machine, verify that the system times on both machines are set the same.

Recommended DBA Maintenance Activities

Apart from daily DBA monitoring and logging activities, the following steps are recommended:

- The queries are tuned to run using a cost-based optimizer. This requires that you analyze the tables and indexes and gather statistics periodically.

- The catalog and auctions module uses the interMedia option for text searches. Text indexes are not automatically updated by Oracle; therefore, it is necessary to manually synchronize the indexes with any changes that have taken place in the actual tables. This synchronization must be performed by the Oracle DBA.

In general, operating on the catalog indexes requires the following steps, where the value PARAM will be replaced by an actual value, depending on the operation:

- a. Log in to the Oracle instance that is hosting the iPlanet Market Maker database as a user for Catalog module.
- b. Run the following:

```
alter index cat.cat_atr_value_txt1 rebuild online parameters
('PARAM');
```

NOTE The single quotes surrounding PARAM are necessary.

In this command, replace 'PARAM' with the following values:

- For synchronization of the indexes, 'PARAM' = sync.
 - For full optimization of the indexes, 'PARAM' = optimize full.
 - For full optimization, but only allowing the optimizer to run for a certain length of time, 'PARAM' = optimize full maxtime N, where N is the time in minutes that the optimizer will run.
- We recommend using the `ctxsrv` process (or `dbms_job` process) to perform synchronization. This process (written and supplied by Oracle) provides regular synchronization updates. Run the following steps:
 - a. Log in, as the Oracle user, to the machine on which the iPlanet Market Maker database will be hosted.
 - b. Run the following command:

```
ctxsrv [-user ctxsys/passwd[@sqlnet_address]]
      [-personality M]
      [-logfile log_name]
      [-sqltrace]

ctxsrv -user ctxsys/ctxsys -personality M -log ctx.log &
```
 - c. To stop the `ctxsrv` process, run the following steps:

- I. Log in, as the Oracle user, to the machine on which the iPlanet Market Maker database will be hosted.
- II. Using SQLPLUS, log into Oracle as user `ctxsys`, password `ctxsys`.
- III. Run the following:

```
begin ctx_adm.shutdown; end;
```

- In addition, the DBA should perform a full optimization of the indexes regularly, depending on the amount of change that has occurred to the catalog.

Run the following command periodically to handle fragmentation of interMedia indexes (running the `ctxsrv` process contributes to index fragmentation.):

```
alter index myindex rebuild online parameters ('optimize fast');
```

or

```
alter index myindex rebuild online parameters ('optimize full');
```

Refer to the Oracle 8.1.7 / 9i interMedia Text Reference for more details on performing periodic maintenance activities.

Recommended DBA Maintenance Activities

Troubleshooting

This chapter discusses problems you may encounter with your system and provides possible solutions. Details on some of these situations can be found in the Chapter 3, “Configuring Your System,” and Chapter 4, “Performance Tuning and Monitoring.”

This chapter addresses the following (alphabetized) error situations:

- Application server and web server would not start.
- Display profile contained bad data.
- E-mail notifications are not sent out.
- kjs process crashes under load.
- LDAP error occur before login.
- New users are not able to log in.
- No available http sessions.
- ORA-29861: domain index is marked LOADING/FAILED/UNUSABLE.
- Oracle exceptions caused by Job Scheduler.
- Oracle JDBC exceptions occur.
- Page flow doesn't correspond with actions.
- Slow login time.
- Slow performance.
- Slow response time.

Use the log files and debugging messages when trying to resolve issues within iPlanet Market Maker.

Application server and web server would not start.

The LDAP server in which the iPlanet Application Server and iPlanet WebServer plug-in are registered has not started.

Solution:

Before starting the iPlanet WebServer or iPlanet Application Server, start the LDAP where they are registered.

Display profile contained bad data.

On invoking the gateway JSP file (`imm.jsp` by default), this error message occurs.

This could be caused by the following:

- Display profile load didn't go through or was not done as a part of the install.
- The Oracle database is not up or the iPlanet Market Maker hosts cannot connect to Oracle database due to improper settings in the `VortexConfiguration.properties`.
- There was an attempt to reload community from scratch overwriting the previous community load without running the display profile load again.

Solution:

- Run the `install.bash` script from the `imm_HOME/tools/etc` directory if this has not been done after the LDAP/Oracle loading with the Installer.
- Check to see if the Oracle instance is up and running and if the settings in the iPlanet Market Maker hosts to connect to Oracle database are made correctly.
- If you are reloading community data, for any reason, overwriting the existing community data, ensure that the `install.bash` is run again. This must be a reload from an LDIF which ensures creation of new guids for all loaded users. This doesn't apply to restorations of community from backups because the GUIDs from the backup are the same as the GUIDs in the database.

E-mail notifications are not sent out.

There is an incorrect SMTP host setting in the file `VortexConfiguration.properties` file.

Solution:

Ensure that the SMTP host setting specified during installation in the file `VortexConfiguration.properties` is correctly specified.

kjs process crashes under load.

JVM is running out of memory.

Solution:

Adjust the maximum heap size of the JVM in the KJS script as mentioned in the performance tuning section. You can modify this in the KJS under the iPlanet Application Server configuration directory.

LDAP error occur before login.

Before logging in for the first time, while accessing the login page, LDAP exceptions are thrown.

The iPlanet Market Maker instance is not connecting to LDAP properly.

Solution:

- Check the settings for LDAP in the `userxconfig.properties` file. Look for the following keys:

```
ldaphost: localhost
ldapport: 389
```

- Check the settings for LDAP in the `VortexConfiguration.properties` file. Look for the following keys:

```
#CFG_LDAP_READ_USER
*10=f2zgvSFyajQORAFrsZXexKebDLWW3cAk
#CFG_LDAP_READ_PASSWORD
*11=iT2NSQTfIUM8zRefAD7VjQ==
#CFG_LDAP_WRITE_USER
*12=f2zgvSFyajQORAFrsZXexKebDLWW3cAk
#CFG_LDAP_WRITE_PASSWORD
*13=iT2NSQTfIUM8zRefAD7VjQ==
```

Make the appropriate adjustments to the settings for your configuration.

Login fails for interoperability.

If your system is configured for interoperability with iPlanet Market Maker and iPlanet BuyerXpert located on separate machines, login will fail if the system time on both machines is not the same.

Solution:

Set the same system time on both machines.

New users are not able to log in.

As the number of users logged in increases beyond a point, new users are not able to log in; an Oracle exception occurs. There are two possible problems.

- The job scheduler is not running.

Solution:

The job scheduler should be running for the e-mail notifications to be sent out. Ensure that it is up and running in its JVM.

- The database connection pool has reached the maximum limit and so is unable to create more connections for the users.

Solution:

Change the database connection pool maximum size by modifying the properties in the `VortexConfiguration.properties` file.

No available http sessions.

Under high load conditions, the web server error log indicates that we have run out of available http sessions.

The maximum number of valid sessions allowed by default in iPlanet WebServer is 1000; the inactive session time out on the server is 30 minutes. If more than 1000 users demand valid sessions from the same iPlanet Market Maker host in a total duration of 30 minutes, you might see the server return an internal server error with the status code of 500, which causes the web server error log to indicate that you have run out of available sessions.

Solution:

Change the `maxSessions` setting in the `<iPlanet-HOME>/webserver-instance/config/context.properties` file to increase the number of maximum allowed valid sessions and/or reduce the session time-out on iPlanet WebServer.

ORA-29861: domain index is marked LOADING/FAILED/UNUSABLE.

The web page throws this Oracle error message. This is an Oracle interMedia index error on the column value of the table `cat_attr`, which means that interMedia index `CAT_ATR_VALUE_TXT1` is in a state of loading. (Also, check Oracle error messages document).

Solution:

The database administrator must drop the index with the force option and rebuild when you get this error. Issue the following commands to fix:

- a. Log into SQLPLUS as cat user.
- b. Issue the following using the force option:

```
"drop index CAT_ATR_VALUE_TXT1 FORCE;"
```

- c. Recreate the index:

```
"create index CAT_ATR_VALUE_TXT1 on CAT_ATTR (VALUE)  
indextype is ctxsys.context;"
```

Oracle exceptions caused by Job Scheduler.

After stopping and restarting Oracle, there are exceptions thrown from Oracle reads by the job scheduler.

Solution:

The job scheduler process needs to be stopped and restarted every time Oracle is shut down and restarted.

Oracle JDBC exceptions occur.

Before logging in for the first time, while accessing the login page, Oracle JDBC exceptions are thrown.

iPlanet Market Maker is not connecting to the Oracle database properly.

Solution:

Check the settings for the Oracle database in the `VortexConfiguration.properties` file. Look for the following keys:

```
#CFG_DATABASE_DRIVER  
# - Specifies the JDBC driver file.  
1=oracle.jdbc.driver.OracleDriver  
  
#CFG_DATABASE_URL
```

```
# - Specifies the database connectivity URL
#Type4 -- 2=jdbc:oracle:oci8:@vortex

#CFG_DATABASE_USER
# - Specifies the database user.
*3=T8bQpW4AQTw=

#CFG_DATABASE_PASSWORD
# - Specifies the database password.
*4=T8bQpW4AQTw=
```

Make the appropriate adjustments to the settings for your configuration.

Page flow doesn't correspond with actions.

In a load-balanced environment, the page flow does not seem to correspond with the actions you follow.

Resonate is not set up correctly to work with the iPlanet Market Maker software.

Solution:

Verify that you have correctly installed Resonate. Important aspects of the installation include:

- Setting up the nodes that represent the iPlanet Market Maker servers that the load will be balanced on.
- Configuring the VIP (Virtual IP) to an IP address in the subnet that is not being used.
- Configuring Resonate for HTTP load balancing to balance among all the iPlanet Market Maker servers on port 80.
- Configuring Resonate for cookie persistence based on the name of the cookie being returned from the server.
- Starting Resonate.
- Configuring each iPlanet WebServer with the iPlanet Market Maker installation to have appropriate (and unique) cookie names for the session.
- Restarting each iPlanet WebServer installation so that the cookie name settings take effect.

Slow login time.

In the display profile of the user, the In-box channel can be set to display the number of emails in the Shared In-box on the portal page every time the user logs in. If this is set, the Shared In-box should be cleaned regularly. If the In-box is not cleaned regularly and if it holds a large number of emails, it could result in longer logging time.

Solution

Delete the emails from the Inbox regularly or remove it as a channel from the users' display profile for the portal page.

Slow performance.

Very slow performance (hours instead of minutes) may occur when performing attribute normalization (an optional operation) for a catalog.

Solution:

Verify that every attribute mentioned in the incoming catalog is contained in any of the translation tables being used. This applies to every version of the iPlanet Market Maker software.

Slow response time.

Exceptionally long response times occur when users try to log in after deployment.

Could be caused by the following:

- Missing indexes in the LDAP settings.
- The community size has exceeded the ALL_IDS_THRESHOLD setting.
- LDAP connection pool is undersized.

Solution:

- Look for any notes=U entries in the LDAP access logs. These entries indicate that there could be a missing index in LDAP for a particular search attribute. For the notes=U entry, identify the attributes in the LDAP search filter and check the slapd.ldbm.conf file in LDAP config settings to see if there is any attribute on which the index has not been created.
- If the community size exceeds the ALL_IDS_THRESHOLD setting, you can expect to see notes=U entries in the LDAP access logs with large etimes (elapsed times). The only solution presently is to adjust the ALL_IDS_THRESHOLD upward.

- Look for the sizing of the LDAP connection pool in the `userxconfig.properties` file. The LDAP clients might be running out of free LDAP connections in the LDAP connection pool. Ensure that, on each iPlanet Market Maker host, the connection pool is set to have minimum and maximum settings of 150 and 300 respectively. These are comfortable settings for expected loads.

Oracle Table Sizing Information

This appendix provides information to help you estimate the extent size for tables used in iPlanet Market Maker. The list in the Table A-1 gives the average row length for each of the tables for a simulation environment conducted using real users.

NOTE The average lengths are based on one installation of iPlanet Market Maker software; the actual average lengths for each installation may vary. Care should be taken to add an additional buffer of 30 percent to account for variations.

The average row lengths are provided to assist in storage calculations for each of the tables. The storage calculations need not be done for small or stagnant tables because they would not have many inserts, deletes, or updates. However, the row lengths can be used to calculate the initial and next extent storage parameters of the core tables.

The following formula can be used to estimate the size of the table in blocks:

$$\text{table size in blocks} = \frac{\text{total_records}}{(((\text{blocksize} - \text{block header}) - (\text{inittrans} * 23)) * ((100 - \text{pctfree}) / 100)) / (\text{average_row_length} + 3 + \text{number of columns} + \text{number of long columns})}$$

Where:

- block_size, and block header are in bytes.
- The inittrans, and pctfree (for example, inittrans = 1; pctfree = 40) are specified as absolute numbers for the table.
- Number of bytes used by inittrans = 23 bytes (used in the above formula).
- Average Row Length = estimated average row size for the table.
- Total Records = total number of records estimated for the table.

Based on the total table size in blocks, you can estimate the initial and next extent sizes.

You can also refer to Tuning Oracle section in Chapter 4, “Performance Tuning and Monitoring,” for a few examples of extents sizes for core tables of iPlanet MarketMaker installation.

Table A-1 can be used as a guide for sizing the extents for each of the Oracle tables in a production environment.

Table A-1 Oracle Table Sizing Information

Table Name	Average Row Length (Bytes)
ATTACHMENTS	121
AUC_ACCESS_LIST	70
AUC_AUCTION	273
AUC_AUCTION_RUNTIME	77
AUC_AUCTION_STATUS	39
AUC_AUCTION_TYPES	90
AUC_BID_HISTORY	210
AUC_FAVORITE	69
AUC_POLICY_ATTR	67
CAT_ATTR	148
CAT_ENUM	n/a
CAT_ENUM_VALUES	n/a
CAT_MASTER	73
CAT_MASTER_ENTRY	130
CAT_SEARCH_ATTR	83
CAT_UNIT	13
CAT_VIEW	106
CAT_VIEW_COMPANY_FILTER	70
CAT_VIEW_ENTRY	179
CAT_VIEW_ROLE_MAP	69
CMN_CONFIGS	770
CMN_CONFIG_PROFILE	649

Table A-1 Oracle Table Sizing Information (*Continued*)

Table Name	Average Row Length (Bytes)
CMN_JOB_SCHEDULER_ENTRY	168
CMN_LOG	138
CMN_LOG_NOTIFY_CONFIG	45
CMN_PLM	59
CMN_PLM_DATE	21
CMN_PLM_INT	16
CMN_PLM_STRING	38
COM_ADDRESSES	121
COM_COMPANIES	193
COM_UPDATE_LOG	83
COM_USERS	216
INBOX_ENTRY	133
MAIL_INFO	329
OMS_APPROVAL_CONFIG	88
OMS_APPROVER	135
OMS_BILL_INFO	102
OMS_BILL_INFO_TEMPLATE	179
OMS_LOCATION_TEMPLATE	n/a
OMS_NOTE	184
OMS_ORDER	387
OMS_ORD_LINE_ITEM	187
OMS_PAYMENT_METHOD_TEMPLATE	112
OMS_PAY_INFO	59
OMS_REQUISITION	308
OMS_REQ_LINE_ITEM	186
OMS_SHIP_INFO	110
OMS_SHIP_INFO_TEMPLATE	182
PRI_EXCH_RATE	34

Table A-1 Oracle Table Sizing Information *(Continued)*

Table Name	Average Row Length (Bytes)
PRI_PRICING_RULE	1197
PRI_RULE_GROUP	103
RFX_REPLIES	292
RFX_REPLIES_ATTS	72
RFX_REPLIES_ATTS_D	n/a
RFX_REPLIES_ATTS_H	71
RFX_REPLIES_D	0
RFX_REPLIES_H	341
RFX_REPLY_ITEMS	224
RFX_REPLY_ITEMS_D	n/a
RFX_REPLY_ITEMS_H	217
RFX_REQUESTS	230
RFX_REQUESTS_ATTS	71
RFX_REQUESTS_ATTS_D	71
RFX_REQUESTS_ATTS_H	70
RFX_REQUESTS_D	230
RFX_REQUESTS_H	250
RFX_REQUEST_ITEMS	201
RFX_REQUEST_ITEMS_D	190
RFX_REQUEST_ITEMS_H	203
RFX_REQUEST_RECIPIENTS	92
RFX_REQUEST_RECIPIENTS_D	75
RFX_REQUEST_RECIPIENTS_H	92
EXC_MEMBER	73
EXC_EXCHANGE_STATUS	76
EXC_ORDER_AXES	54
EXC_ORDER_REQUESTS	239
EXC_ORDERS_TEMP	45

Table A-1 Oracle Table Sizing Information (*Continued*)

Table Name	Average Row Length (Bytes)
EXC_ORDER_REQUEST_CRITERIA	17
EXC_ORDER_REQUEST_LIFESPAN	19
EXC_TRADE_MASTER	51
EXC_TRADE_UNITS	77
EXC_TRADE_UNIT_AXES	12
EXC_QUOTE	35
EXC_NVP_EXT_PASSIVE_ORDER	49
EXC_NVP_EXT_PASSIVE_AXIS	54
EXC_NVP_EXT_ACTIVE_ORDER	n/a
EXC_NVP_EXT_ACTIVE_AXIS	n/a
EXC_NVP_EXT_TEMP_MATCHES	n/a
EXC_NVP_EXT_MATCH_RESULTS	n/a
EXC_COMPANY_ACCOUNT	88
EXC_TRADER_ACCOUNT	69
EXC_ORDER_AUDIT_LOG	127
EXC_COMPANY_PREFER	73
EXC_SYMBOL_STATUS	21
EXC_PROCESSED_MESSAGES	29
EXC_PUB_MESSAGES_M	n/a
EXC_PUB_MESSAGES_D	n/a
EXC_PUB_STATUS	6
EXC_MATCHING_ENGINE_LOCK	11
EXC_NVP_PASSIVE_ORDER	49
EXC_NVP_PASSIVE_AXIS	54
EXC_NVP_ACTIVE_ORDER	n/a
EXC_NVP_ACTIVE_AXIS	n/a
EXC_NVP_TEMP_MATCHES	n/a
EXC_NVP_MATCH_RESULTS	n/a

Table A-1 Oracle Table Sizing Information (*Continued*)

Table Name	Average Row Length (Bytes)
EXC_ORDER_SCHEDULER	57
EXC_ORDERS	103

NUMERICS

7X24 solution 30

A

advanced development setup 26
ALL_IDS_THRESHOLD parameter 78, 107
architecture 17
attribute normalization 107
AUC_AUCTION table 56
AUC_BID_HISTORY table 56
auction-intensive marketplace 44
Auctions module
 table sizes 56
audience for this guide 11
availability requirements 27

B

basic development setup 26
basic production setup 28

C

cache tuning 76
calculator for sizing 41
CAT_ATTR table 55
CAT_MASTER_ENTRY table 55
CAT_SEARCH_ATTR table 55, 56
CAT_VIEW_ENTRY table 55
Catalog module
 attribute normalization issue 107
 tables 55
checklist for deployment 20
CMN_PLM table 57
CMN_PLM_STRING table 57
concurrent users 27, 51
configuring
 disks 72
 iPlanet Directory Server (LDAP) 50
 iPlanet Market Maker 49
 Net8 49
 Oracle 47
configuring your system 47–64
connection pool settings 51, 107
cookies 106
CPU requirements 42, 43
CPU selection 31
ctxsrv process 96, 98, 99

D

database

- cache tuning 76
- reinstalling 54
- script logs 53
- sizing 28

database cache (LDAP) 74

db_block_buffers parameter 48

db_block_lru_latches parameter 48

db_home_directory 75

DBA activities 97, 99

dbms_job process 98

delayed callbacks 60

deployment

- checklist 20
- plan 25
- verification 95

development setup 25

disks

- array striping 73
- configuring 72
- selection 31

display profile 107

documentation 12

E

e-mail notifications 104

entry cache tuning 74

error logs 92

error situations 101–108

exchange-intensive marketplace 46

execution log 53

extents 56, 57

extents sizing 110

F

firewalls 97

fragmentation of indexes 99

G

gateway JSP file 102

H

hard disk space 54

hardware

- components 31
- configurations 30
- requirements 28, 41

heap size 89

high-availability production setup 29

HOST command 83

http sessions 104

I

I/O distribution 73

I/O statistics tool 90

iascontrol kill command 89

iascontrol start command 89

iasdeploy command 87

ias-web.xml file 87

imm.jsp 87

In-box channel 107

in-box notifications 57

INBOX_ENTRY table 57

index

- fragmentation 99
- statements 53
- VLV 82

index updating 98

initialization parameters 47, 49, 96

installation

- Oracle settings 52

- planning 22
- Installer 52, 53, 54
- installParams.env file 52
- installsql.sh script 52, 53
- interMedia 98, 99, 105
- interoperability 97, 104
- iostat command 90
- iPlanet
 - customer support 14
- iPlanet Application Server
 - logs 92
 - session model 86
 - tuning 84
 - verification 97
- iPlanet Directory Server
 - configuring 50
 - indexes 78
 - logs 93
 - tuning 73
 - verification 96
- iPlanet Market Maker
 - architecture 17
 - configuring 49
 - logs 91
 - overview 17
 - tuning 89
- iPlanet Professional Services 31
- iPlanet WebServer
 - logs 92

J

- Java heap size setting 89
- Java version 22
- JDK patches 23
- job scheduler 58, 97, 104, 105
- JVM heap 72, 103

K

- kernel parameters (Solaris) tuning 70
- KJS processes 85
- kregedit command 88, 97
- ksvradm command 85

L

- LDAP
 - access logs 107
 - connection pool 107
 - connection pool settings 51
 - indexes 78
 - remote instances 50
 - schema 50
 - transaction logs 77
 - tuning 73
- ldaphost 50, 96
- ldapport 50, 96
- LDIF 81, 102
- load balancing 76, 87, 88, 106
- log files 91

M

- marketmaker requirements 27
- marketplace events 58
- maximum heap size for the JVM 58
- maxSessions setting 104
- memory tuning 71
- monitoring performance 90
- monitoring tools 90
- mpstat command 91

N

- Net8 49, 96

- network activity monitoring 91
- ns-httpd 85
- ns-slaped 81, 85

O

- OnceOnly database polling thread 60
- OnceOnly timer thread 58
- OnceOnly worker threads 59
- optimizer for Oracle 97
- Oracle
 - configuring 47
 - connecting remote instances to LDAP 51
 - connection pool settings 51
 - errors 105
 - installation settings 52
 - maintenance 99
 - memory requirements 72
 - optimizer 97
 - table sizing 109
 - tuning 82
 - verification 96
- overview
 - deployment 20
 - iPlanet Market Maker 17

P

- page flow error 106
- patches 23, 24
- performance monitoring 90
- performance tests 41, 69
- performance tuning 69–93
- planning deployment 17–39
- pricing engine, turning off 89
- process statistics 90
- processor statistics 91
- production setup 28
- production setup with SSL 29
- properties for the job scheduler 58

- proxy servers 71, 97
- prstat command 91

R

- RAM
 - selection 31
 - tuning 71
- read performance for LDLAP 73
- reg.dat file 87
- repeatable requests from timers 59
- Repeatable thread pool size 59
- Repeatable timer thread 58
- requirements
 - for the marketmaker 27
 - hardware 28
 - memory 71
- Resonate configuration 106
- response time 42
- response times for LDAP 75
- RFx marketplace 45
- RFx module 59
- round robin load balancing 88

S

- schema
 - configuring 51
 - creation 52
 - recreating 54
- security 27
- session model for iPlanet Application Server 86
- session time-out on iPlanet WebServer 104
- Shared In-Box 58, 107
- simultaneous users 27
- sizing calculator 41
- sizing parameters for tables 54
- sizing your system 41–46
- slapd.conf file 77

- slapd.ldbm.conf file 77, 79, 80, 96
- SMTP host setting 103
- snoop command 91
- Solaris
 - tuning kernel parameters 70
 - verification 95
- sorting 48
- SQL scripts 53, 54
- SQLPLUS 49, 55, 96, 105
- SSL
 - decryption 27, 71
 - production setup 29
- static server 71
- statistics for monitoring 90
- Sticky Load Balance 87, 97
- storage parameters 54
- striped logical disks 73
- Sun Cluster 29
- synchronization (Oracle) 98
- System Global Area (SGA) 47
- system tuning 70
- System_JSPRunner 87, 97
- System_JSPRunnerSticky 87, 97

T

- tables
 - Auctions module 56
 - Catalog module 55
 - Exchange module 56
 - in-box notifications 57
 - sizing 109
- tablespaces
 - creation 53
 - distributing 82
- TEMP tablespace 53
- tests 69
- threads for job scheduler 58
- tools for monitoring 90
- transaction log files 77
- Transactional Mail Scheduler 59

- troubleshooting 101–108
- tuning 69–93
 - iPlanet Application Server 84
 - iPlanet Directory Server 73
 - iPlanet Market Maker 89
 - Oracle 82

U

- usage expectations 27
- USERS tablespace 83
- userxconfig.properties file 50, 103, 108

V

- verifying deployment 95
- VIP (Virtual IP) configuration 106
- Virtual List View (VLV) indexing 82
- virtual memory statistics tool 90
- vmstat command 72, 90
- VORTEX_SID parameter 55
- VortexConfiguration.properties file 50, 51, 58, 103, 104, 105

W

- web.xml file 87
- website
 - for iPlanet customer support 14
- write load distributing 76
- write performance tuning (LDAP) 74

