Novell exteNd Application Server

5.2

www.novell.com

ADMINISTRATOR'S GUIDE

Novell®

Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to makes changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

This product may require export authorization from the U.S. Department of Commerce prior to exporting from the U.S. or Canada.

Title to the Software and its documentation, and patents, copyrights and all other property rights applicable thereto, shall at all times remain solely and exclusively with SilverStream and its licensors, and you shall not take any action inconsistent with such title. The Software is protected by copyright laws and international treaty provisions. You shall not remove any copyright notices or other proprietary notices from the Software or its documentation, and you must reproduce such notices on all copies or extracts of the Software or its documentation. You do not acquire any rights of ownership in the Software.

Patent pending.

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.

www.novell.com

exteNd Application Server*Administrator's Guide*
June 2004

Online Documentation: To access the online documemntation for this and other Novell products, and to get updates, see www.novell.com/documentation.

Novell Trademarks

ConsoleOne is a registered trademark of Novell, Inc.

eDirectory is a trademark of Novell, Inc.

GroupWise is a registered trademark of Novell, Inc.

exteNd is a trademark of Novell, Inc.

exteNd Composer is a trademark of Novell, Inc.

exteNd Director is a trademark of Novell, Inc.

iChain is a registered trademark of Novell, Inc.

jBroker is a trademark of Novell, Inc.

NetWare is a registered trademark of Novell, Inc.

Novell is a registered trademark of Novell, Inc.

Novell eGuide is a trademark of Novell, Inc.

SilverStream Trademarks

SilverStream is a registered trademark of SilverStream Software, LLC.

Third-Party Trademarks

All third-party trademarks are the property of their respective owners.

Third-Party Software Legal Notices

The Apache Software License, Version 1.1

Copyright (c) 2000 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Apache" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

JDOM.JAR

Copyright (C) 2000-2002 Brett McLaughlin & Jason Hunter. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution. 3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact license@jdom.org. 4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management (pm@jdom.org).

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following: "This product includes software developed by the JDOM Project (http://www.jdom.org/)." Alternatively, the acknowledgment may be graphical using the logos available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Sun

Sun Microsystems, Inc. Sun, Sun Microsystems, the Sun Logo Sun, the Sun logo, Sun Microsystems, JavaBeans, Enterprise JavaBeans, JavaServer

Pages, Java Naming and Directory Interface, JDK, JDBC, Java, HotJava, HotJava Views, Visual Java, Solaris, NEO, Joe, Netra, NFS, ONC, ONC+, OpenWindows, PC-NFS, SNM, SunNet Manager, Solaris sunburst design, Solstice, SunCore, SolarNet, SunWeb, Sun Workstation, The Network Is The Computer, ToolTalk, Ultra, Ultracomputing, Ultraserver, Where The Network Is Going, SunWorkShop, XView, Java WorkShop, the Java Coffee Cup logo, Visual Java, and NetBeans are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Indiana University Extreme! Lab Software License

Version 1.1.1

Copyright (c) 2002 Extreme! Lab, Indiana University. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Indiana University Extreme! Lab (http://www.extreme.indiana.edu/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Indiana University" and "Indiana University Extreme! Lab" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact http://www.extreme.indiana.edu/. 5. Products derived from this software may not use "Indiana University" name nor may "Indiana University" appear in their name, without prior written permission of the Indiana University.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS, COPYRIGHT HOLDERS OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Phaos

This Software is derived in part from the SSLavaTM Toolkit, which is Copyright ©1996-1998 by Phaos Technology Corporation. All Rights Reserved. Customer is prohibited from accessing the functionality of the Phaos software.

W3C

W3C® SOFTWARE NOTICE AND LICENSE

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications: 1.The full text of this NOTICE in a location viewable to users of the redistributed or derivative work. 2.Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code. 3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

Contents

About This Book

Purpose

This book explains how to administer the Novell® exteNd™ Application Server.

Audience

This book is for the Novell exteNd Application Server administrator.

Organization

This table provides a brief summary of the book's contents.

Part or chapter	Contents
Chapter 1, "Administration Quick Reference"	A listing of cross-references to help you get to the information you want as fast as possible.
Part I, "Administration Basics"	Introduces the Novell exteNd Application Server's three-tiered architecture and outlines the system administrator's tasks. It also introduces the application server's Server Management Console (SMC), which you will use to perform many administrative tasks. Describes the basic hardware configurations for the Novell exteNd Application Server and explains how the server operates in the Web environment. Describes configuration settings for data sources and describes the SilverMaster database.
Part II, "Administering the Server"	Provides instructions for common administrative tasks that you will need to perform, including: ◆ Running the server, including how to start and stop the server and how to log server activity ◆ Setting up Silver Security users and groups ◆ Performing common maintenance tasks such as adding databases to the server and managing deployed applications ◆ Using Web Server Integration (WSI) modules to integrate your application server with an external Web server ◆ Setting up and using security in both the HTTP and HTTPS environments ◆ Tuning the application server for optimum performance, including managing connections to clients and connection pools ◆ Administering a cluster to provide load balancing and failover ◆ Using the Server Administration API to programmatically administer the application server ◆ Troubleshooting

Part or chapter	Contents
Part III, “Appendixes”	Describes the httpd.props file, which you can use to edit selected server properties. Describes how Simple Network Management Protocol (SNMP) is implemented, and how to run SNMP on your application server. Describes the application server’s system tables and SilverMaster database URLs. Describes the directives and configuration settings for the Web Server Integration (WSI) modules.

Additional documentation

📖 For the complete set of Novell exteNd documentation, see the Novell documentation Web site (www.novell.com/documentation/exteNd.html).

1 Administration Quick Reference

Use this Quick Reference to quickly get to the following information:

- SMC panels
- Administration tasks

SMC panels

The SMC is divided into these areas:

- Configuration options
- Security options
- Monitor options
- Deployment options

This section describes the panels in each of the areas.

NOTE: The panels are different if you are managing a clustered environment. For a quick reference to the SMC panels for a cluster, see "Administering a server cluster" on page 209.

Configuration options

Configuration options consist of the following panels:

Panel	Description / Where to go for more information
General	General, server logging, and ORB/RMI settings. Use this panel to configure separate ports for different types of users and operations. 📖 See: ◆ **General**: "Specifying general server properties" on page 80 ◆ **Ports**: "Setting up separate ports" on page 79 ◆ **Logging**: "Using server logging" on page 82 ◆ **ORB/RMI**: "Specifying ORB settings" on page 84
Advanced	Debugging, performance, server cache, and J2EE transactions. 📖 See: ◆ **Debug**: "Low-level debugging" on page 219 ◆ **Performance**: "Setting performance parameters" on page 181 ◆ **Cache**: "Managing the server content cache" on page 187 ◆ **Transactions**: "Managing J2EE transactions" on page 99

Panel	Description / Where to go for more information
Pools	Adding, removing, and maintaining JDBC and Connector connection pools. See: ◆ **Managing JDBC connection pools**: “Configuring connection pools” on page 51 ◆ **Managing Connector connection pools**: “Configuring connection pools” on page 51
Databases	Adding and removing databases. Configuring how the server accesses the SilverMaster database and deployment databases. You can also use this panel to synchronize database schema and delete idle connections. See: ◆ **Adding and remove databases**: “Configuring deployment databases” on page 46 ◆ **Minimum and maximum number of connections**: “Managing database connections” on page 192 ◆ **All other configuration tasks**: “Configuring deployment databases” on page 46
Connections	Managing client connection settings. See “Client connection parameters” on page 183

Security options

Security options consist of the following panels:

Panel	Description / Where to go for more information
General	General security settings. See: ◆ **Require user authentication, Disable HTML directory listing, and Allow users to modify own account**: “Enabling authentication” on page 159 ◆ **Security resource timeout**: “Resetting the security resource timeout” on page 126 ◆ **Default security realm and authority**: “Overriding defaults for login name components” on page 134
Advanced	HTTPS client certificate levels, accelerator settings, and trusted clients. See: ◆ **Client certificate level in HTTPS**: “Enabling and installing client certificates” on page 153 ◆ **Accelerator settings**: “Using Cryptographic Hardware Integration” on page 161 ◆ **List of trusted clients**: “Managing trusted clients” on page 161
Permissions	You can set up access control at the cluster, server, or database level. To control access to J2EE archives, you can set up access control on the Deployed Objects directory of the database where these objects are deployed. See “Authorization and access control” on page 165

Panel	Description / Where to go for more information
Users & Groups	Adding Silver Security and certificate users. Adding Silver Security groups. Viewing users in external security providers. Editing user properties. 📖 See: ◆ **Adding Silver Security users and groups**: “Managing Silver Security users and groups” on page 90 ◆ **Adding certificate users**: “Manually installing client certificates” on page 155 ◆ **Viewing users and groups**: “Accessing users and groups” on page 132 ◆ **Using external security providers**: “Accessing security provider systems” on page 124 ◆ **Editing user properties**: “Editing user properties” on page 91
Certificates	Viewing certificates that have been installed on the server. Viewing recognized Certificate Authorities. 📖 See: ◆ **Creating and installing server certificates**: “Creating and installing server certificates using the SMC” on page 138 ◆ **Viewing server certificates**: “Viewing server certificates” on page 149 ◆ **Managing Certificate Authorities**: “Managing Certificate Authorities” on page 153 ◆ **DSA and RSA port properties**: “Enabling RSA/DSA ports” on page 149
Security Providers	Configuring the application server to recognize external security providers, including Windows directory services, LDAP, NIS+, and certificate issuers. 📖 See “Accessing security provider systems” on page 124.

Monitor options

Monitor options consist of the following panels:

Panel	Description / Where to go for more information
Charts	Displaying real-time charts of various server statistics. 📖 See “Displaying charts of server activity” on page 100
Logs	Displaying logs if you have enabled server logging. 📖 See “Displaying logs” on page 103
Statistics	Displaying tabular views of server statistics related to sessions and threads, as well as summary statistics. 📖 See “Displaying views of server statistics” on page 104

Deployment options

Deployment options consist of the following panels:

Panel	Description / Where to go for more information
Deployed objects	Viewing and managing J2EE objects. Use this panel to enable, disable, and shut down J2EE applications that have been deployed on the server. 📖 See “Maintaining deployed J2EE objects” on page 96
JNDI tree	📖 See “Maintaining deployed J2EE objects” on page 96
Manage URLs	📖 See “Maintaining deployed J2EE objects” on page 96
Resource Adapters	📖 See “Maintaining deployed J2EE objects” on page 96

Administration tasks

This section provides a quick reference to common administration tasks.

Data source configuration

- “Configuring deployment databases” on page 46
- “Configuring connection pools” on page 51

General server management

- “Starting the application server” on page 69
- “Shutting down the application server” on page 74
- “Restarting the application server” on page 74
- “Setting up separate ports” on page 79
- “Specifying general server properties” on page 80
- “Using server logging” on page 82
- “Specifying ORB settings” on page 84
- “Running multiple servers on one host” on page 86
- “Managing Silver Security users and groups” on page 90
- “Administering an application server remotely” on page 95
- “Setting the AGCLASSPATH variable” on page 96

Security

- “Establishing a secure connection to the server” on page 122
- “Accessing security provider systems” on page 124
- “Using certificates” on page 136
- “Enabling RSA/DSA ports” on page 149
- “Enabling authentication” on page 159
- “Restricting permissions” on page 169
- “Excluding robots” on page 180
- “Locking down servers, clusters, and applications” on page 173
- “Security checklist” on page 174

Tuning and performance

Load balancing and failover

Troubleshooting

I Administration Basics

This part describes the basics of administering the Novell exteNd Application Server

- Chapter 2, “Administration Overview”
- Chapter 3, “Server Configuration”
- Chapter 4, “Data Source Configuration”

2 Administration Overview

This chapter introduces the Novell exteNd Application Server architecture and outlines administrative tasks in the server environment. It contains sections on:

- The Novell exteNd Application Server
- Application server administration
- The Server Management Console (SMC)

The Novell exteNd Application Server

The Novell exteNd Application Server is a multithreaded J2EE application server implemented in Java. Client communications are conducted through the HyperText Transfer Protocol (HTTP), the most common protocol for the World Wide Web.

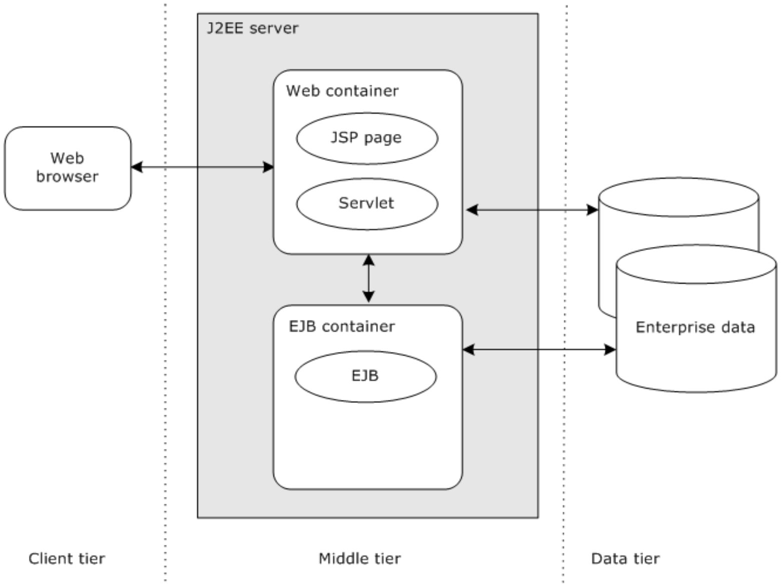
NOTE: The application server uses RMI (Java's Remote Method Invocation) instead of HTTP when EJBs on different servers communicate.

The application server provides business logic processing and access to corporate data.

Three-tiered communications

The application server supports a three-tiered architecture that consists of a **client tier**, a **middle tier**, and a **data tier**.

Tier	Description
Client	Web browsers or standalone application clients.
Middle	Includes the Novell exteNd Application Server. The middle tier includes two runtime environments: ◆ **Web container**—Provides support for receiving and responding to client requests ◆ **EJB container** (or business tier)—Where the business logic (including data access) resides
Data	Includes corporate data accessed via relational databases or J2EE Connectors. For a list of supported databases and connectors, see the *Release Notes*.



Three-tiered communications provide the following benefits:

Benefit	Description
Security management	The application server mediates all communication to the data tier, enforcing the security you set up.
Code management	By encapsulating your business logic into server-side objects, it is easier to manage and maintain code, especially in a large-scale development environment.
Data validation	Because the business logic is contained in one tier, you can protect data by controlling access and operations from one central point.

Application server environments

As an application server administrator, you'll set up and support these environments:

Environment	Description
Production	The production environment consists of one or more application servers, one or more database servers or Enterprise Information Systems (EIS), and clients. 📖 For more information about the production environment, see Chapter 3, "Server Configuration".
Deployment	The deployment environment consists of one or more application servers, one or more database servers or EIS systems, and the deployment tools. ◆ Deployment responsibilities can include mapping role references to users and groups in the security system and mapping resource references to data sources. ◆ Deployment tools include SilverCmd or the deployment tools provided by Novell exteNd Director™. 📖 For more information about deployment, see the chapter on J2EE archive deployment in the *Facilities Guide*.

Application server administration

As the application server administrator, you have some administrative responsibilities for each tier in the server architecture: the client tier, the middle tier, and the data tier.

Client tier administration

These are the requirements for running each client type:

Client	Requirements
Web clients	Browser requirements depend on the kind of HTML applications being run. NOTE: Browser administration is not covered in this guide. For more information, see your browser documentation.
Java clients	SilverJ2EEClient is used to host J2EE application clients on user machines. 📖 For more information, see the chapter on SilverJ2EEClient in the *Facilities Guide*.

Middle tier administration

When you administer the middle tier, you administer the application server. You use the server's Server Management Console (SMC), a standalone administration tool described later in this chapter.

These are the major areas of application server administration:

Administration area	Description	For information
Installation	Use the Novell exteNd installation program to install the application server.	For instructions, see *Installing Novell exteNd* For the latest system requirements, see the exteNd Application Server *Release Notes*
Access to corporate data sources	You need to create and maintain the connection pools that the server will use to provide access to the corporate data on the data tier.	See Chapter 4, "Data Source Configuration"
Deployment	J2EE applications are deployed to relational databases that have been added to the application server. You may need to add the databases to the server and maintain and tune connections.	See Chapter 4, "Data Source Configuration"
Statistics	Once a production application server is up and running, you can monitor statistics in order to tune performance and schedule maintenance activity.	See Chapter 7, "Maintaining the Server"
Logging	The application server can log different types of system information to either a database or a file.	See "Using server logging" on page 82

Administration area	Description	For information
Certificates	Certificates are used in Secure Sockets Layer (SSL) connections for the server to authenticate itself to clients and for the clients to authenticate themselves to the server. You can install RSA and DSA certificates on the application server.	See Chapter 9, “Setting Up Security”
Security	The application server offers several levels of security.	See Chapter 10, “Using Security”
Server performance	The application server has several settings that define its behavior when busy or under light load. You can define the number of connections required for the application server to be in a specific state.	See Chapter 11, “Tuning the Server”
Load balancing	Load balancing lets you use multiple applications servers (clusters) in a large-scale production environment.	See Chapter 12, “Administering a Cluster”
Troubleshooting	—	See Chapter 14, “Troubleshooting”

📖 For a quick reference of administration tasks, see “Administration tasks” on page 16.

Data tier administration

The application server relies on components located on the data tier for system management resources (the SilverMaster database), deployment targets (deployment databases), and corporate data (connection pools). You’ll need to work with the administrators responsible for these resources to ensure that the application server has the appropriate access.

📖 For more information on how the application server uses resources on the data tier and the administrative implications, see Chapter 4, “Data Source Configuration”.

The Server Management Console (SMC)

The application server’s Server Management Console (SMC) is the tool you use for most application server administration tasks. You can use the SMC to:

- Maintain the environment
- Monitor the environment
- Measure and improve performance
- Set up and administer security
- Set up and maintain server clusters for load balancing

You can administer multiple servers from the same SMC.

SMC or httpd.props file? A few of the SMC settings affect entries in the httpd.props file, which you can edit directly. But whenever possible, use the SMC to change the server’s settings.

📖 For more information about the httpd.props file, see Appendix A, “The httpd.props File”.

Running the SMC

You can run the SMC from the command prompt (system console) or from a GUI.

➢ **To run the SMC from a GUI:**

Operating system	Action/Description
NetWare®	From the GUI screen, choose **Novell>exteNd Application Server>SMC.**
Windows	From the Start menu, choose **Programs>Novell exteNd *n.n*>AppServer>Server Management Console**. If you change the port your server is listening on from the port you installed the server on, you need to update your program shortcut used to launch the SMC.

➢ **To run the SMC from the command prompt or system console:**

- Type the following command (located in the server's **\bin** directory):

```
smc
```

The **smc** command can take the following command-line options:

Command-line option	Description
-as_noconsole	Suppress the Java console at startup
-as_username *username*	Log in using the specified user name
-as_password *password*	Log in using the specified password
-as_nosplash	Do not display the SMC splash screen
+Dsssw.ssl.nocacheck	Do not verify self-signed server certificates
-?or -help	List options

Using ports The application server supports separate **runtime** and **administration** ports. During installation, both HTTP ports are configured to whatever port number you specified as the default. The default ports are:

Operating system	Default port
NetWare	83
UNIX	8080
Windows	80

If you have configured separate server ports, you must specify your administration port number when starting the SMC.

📖 For more information, see "About enabling ports" on page 79.

Creating a secure connection You can establish a secure (SSL) connection between the SMC and the application server. For information, see "Establishing a secure connection to the server" on page 122.

The SMC properties file The SMC properties file (smc.props located in the server's **\Resources** directory) contains information about:

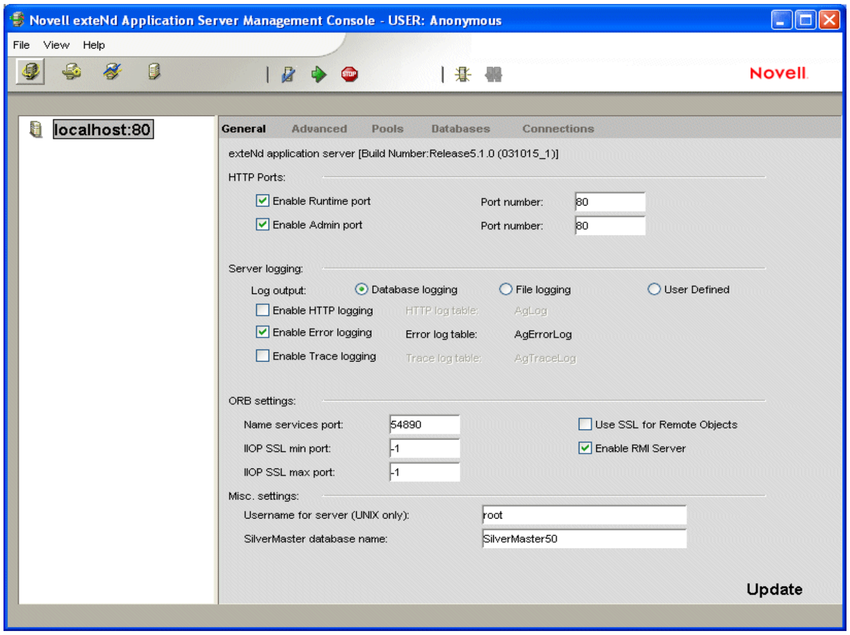
- The list of servers that have been added to the SMC (through the SMC)
- The property that specifies the settings for charting

If you pass a server name on the command line, it is **not** added to the list of servers. If you supply the user name and password on the command line but not the server, the command is ignored (since the SMC cannot determine which server the parameters apply to).

The smc.props file is updated when you use the SMC to make changes to these properties and when you close the SMC. Do not edit smc.props manually while the SMC is running—none of the changes will be saved.

The SMC user interface

The SMC consists of a series of panels that you can use to administer the server:



exteNd application server [Build Number:Release5.1.0 (031015_1)]

HTTP Ports:

☑ Enable Runtime port   Port number: 80

☑ Enable Admin port   Port number: 80

Server logging:

Log output: ● Database logging ○ File logging ○ User Defined

□ Enable HTTP logging   HTTP log table: AgLog

☑ Enable Error logging   Error log table: AgErrorLog

□ Enable Trace logging   Trace log table: AgTraceLog

ORB settings:

Name services port: 54890   □ Use SSL for Remote Objects

IIOP SSL min port: -1   ☑ Enable RMI Server

IIOP SSL max port: -1

Misc. settings:

Username for server (UNIX only): root

SilverMaster database name: SilverMaster50

NOTE: The SMC displays different options if you are running the server in a clustered environment. For more information, see Chapter 12, “Administering a Cluster”.

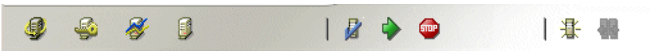
About SMC panels

The administrative options are grouped into panels, such as General, Advanced, and so on.

📖 For a quick reference to the SMC panels, see “SMC panels” on page 13.

About the toolbar

The toolbar at the top of the console displays icons that allow you to perform actions:



Task	Icon	Description
Configuration		Provides access to configuration options—such as general server options, database options, and client connection options
Security		Provides access to security options—such as users and groups, the use of user authentication, certificates, and security providers
Monitor		Provides access to charts of server statistics and logs

Task	Icon	Description
Deployment		Provides access to J2EE objects deployed on the server, the server’s JNDI (Java Naming and Directory Interface) tree, settings for the server’s or database’s default URL, and listings of RARs deployed to the server
Choose (server)		Adds a server on your network to administer using the SMC; you can administer multiple servers from one SMC console
Restart (server)	➡	Restarts the selected server after changing parameters
Stop (server)	STOP	Shuts down the selected server
New (cluster)		Creates a cluster for load balancing
Dissolve (cluster)		Dissolves a load balancing server cluster (applies to server clustering only)

Menu

The menu at the top of the console provides another way to perform many of the same functions that the toolbar provides. It also lets you perform these additional tasks:

Menu option	Description
File>Login	Allows you to log in to the SMC. For more information, see “Logging in” on page 27.
View>Server console	Displays the server console.

Logging in

You must start the application server before you can log in to the SMC.

If you start the SMC without providing a user name or password, you are connected as Anonymous. You can connect as Anonymous only if the application server was installed in unrestricted mode. If the application server was installed in restricted mode (the installation default and the recommended mode for production environments), all users are required to log in.

The SMC uses the admin port for all actions.

➤ **To log in:**

1 Select **File>Login**.

The Enter Login Credentials dialog displays.

NOTE: The application server installs a predefined group named Administrators, which initially contains only the server administrator.

2 Enter your application server administrator user name and password, then click **OK**.

Your administration account name and password are whatever you specified when you installed the application server. Passwords are always case-sensitive. The user name might be case-sensitive if your SilverMaster database is set up to support case-sensitivity. For more information, see “About your administrator account” on page 90.

You now have all administration permissions. The SMC shows the name of the user in its window title.

Logging out

➢ **To log out:**

1. In the left panel of the SMC, select the server.
2. Select **File>Logout**.

   You are now connected to that server as Anonymous (as shown in the window title). If you want, you can log back in as a user.

Online help

To access the administration documentation in the application server's help:

From here	Do this
SMC	Press **F1** or select **Help>Help Topics**. The "Administration Quick Reference" displays in your browser. From there you can access the entire *Administrator's Guide* and the rest of the exteNd Application Server help.
Windows	From the Start menu, select **Programs>Novell exteNd *n.n*>Product Documentation**. Then go to the exteNd Application Server help and open the *Administrator's Guide*.

📖 For more information, see Using Help and Documentation.

3 Server Configuration

This chapter describes basic hardware configurations for the Novell exteNd Application Server and explains how the server operates in the Web environment. It contains sections on:

- Server configurations
- Firewalls and proxy servers
- Network configurations
- Session management

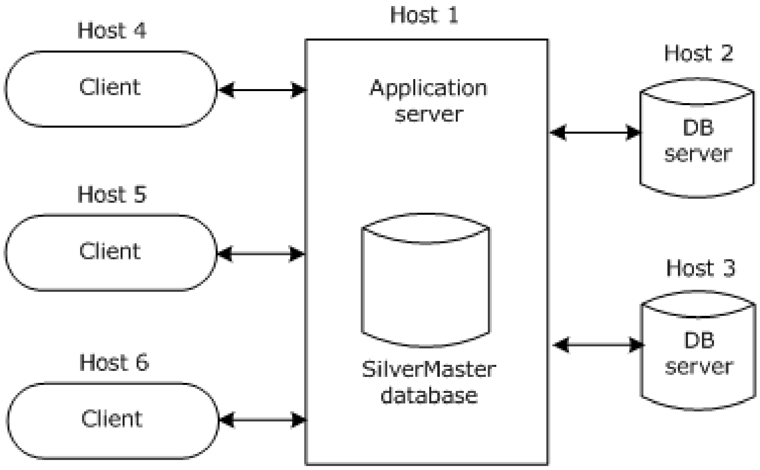
Server configurations

This section describes the recommended application server configurations for production. For simplicity, the descriptions assume a single (standalone) application server.

Production environment

In a production environment, it is best to configure your application server and database server(s) on separate machines. (This is called a *multiple-host configuration*.)

The figure below shows the preferred application server configuration with two database server connections:



The SilverMaster database (shown above with the application server) is a master database for the entire system. For a description of SilverMaster, see “SilverMaster functions” on page 42.

Having another Web server in this configuration would have little impact on the application server. The application server can coexist with Web servers as long as you change the application server’s listening port from the default (port 80) to another port. For more information, see “Specifying general server properties” on page 80.

Benefits Configuring the application server and database servers on separate machines results in the following benefits:

- The application server does not compete with the database servers for CPU and memory resources.
- The machine that hosts each database server can be configured to match that application server's memory requirements.
- Database servers can be optimized and tuned without affecting the application server.
- You can run your database servers on operating system platforms other than the one the application server is running on. For example, you can run UNIX database servers and the application server on Windows.

Drawback One drawback to configuring application servers and database servers on separate machines is that you must maintain extra machines.

📖 For more information about possible configurations for your production application server environment, see "Network configurations" on page 31.

Firewalls and proxy servers

Firewalls are critical for regulating network access. You must make many decisions about how you will use firewalls, how the application server will communicate with database servers, and what if any access you will allow Anonymous users through the firewall.

In a typical large-scale Web environment, a static traffic routing service is placed between the network service provider's router and the internal network. The traffic routing service may be implemented at an IP level using screening rules in a router—or at an application level using proxy gateways and services.

About proxy servers A *proxy server* is an application that mediates traffic between a protected network and the Internet. Proxy servers are used primarily to consolidate Internet connections, provide users a general level of anonymity (by shielding information normally passed from the browser to the Web server), and enforce enhanced security about Web traffic (such as what sites users can access).

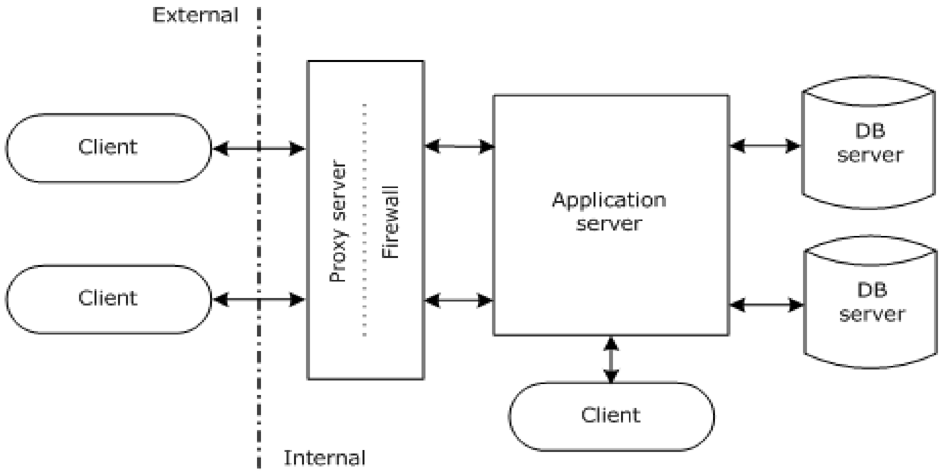
Many proxies contain extra logging or support for user authentication. Since proxies must understand the application protocol being used, they can also implement protocol-specific security. The proxy machine provides a higher level of audit and security, but it also increases configuration costs and reduces the level of service—because a proxy needs to be developed for each desired service.

NOTE: The proxy server software you use with the application server should support HTTP 1.1, such as the Microsoft Proxy Server or the Netscape Proxy Server.

About firewalls A *firewall* is a hardware or software facility used to regulate access to a network. Firewalls are traditionally used to protect the company's intranet from public Internet traffic. *Policies* are configured on the firewall to allow only certain traffic to pass through. The actual mechanism involved varies, but in principle the firewall can be thought of as two mechanisms: one that exists to block traffic and another that exists to permit traffic. Administrators can configure a firewall to notify them of security breaches and monitor overall traffic.

Configuration with a firewall and proxy server

The application server should run **inside** any firewalls your site has, with the HTTP requests from extranet customers to the application server either allowed through the firewall or proxied. This way the database connections need not go through the firewall. The figure below shows how an application server might be configured with a firewall and proxy server:



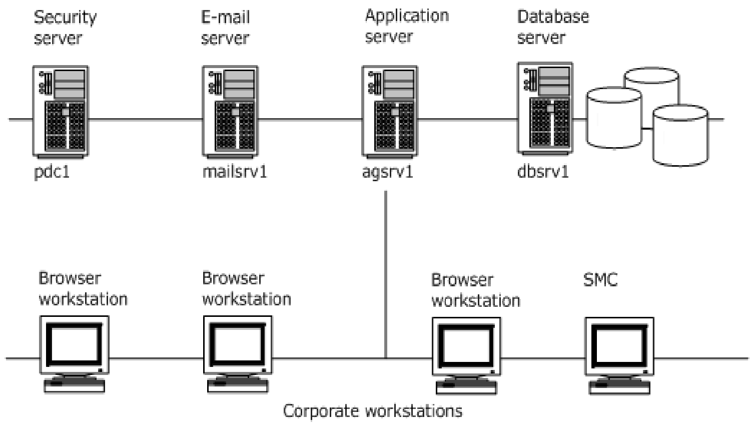
Network configurations

This section presents several possible ways to configure your network based on your application's needs and includes these sections:

- Simple intranet configuration
- Intranet cluster configuration
- Simple Internet configuration
- Internet cluster configuration
- Demilitarized Zone (DMZ) Internet configuration

Simple intranet configuration

Small companies, departments, and small teams of developers can work against a single application server. The following figure shows a simple network configuration with the application server (**agsrv1**) hosting a simple Web application serving users on a local area network (LAN). The application server leverages an existing Windows security domain (on **pdc1**) and e-mail server (**mailsrv1**) for user authentication/access control and pushing application data to users via e-mail:



The server's SilverMaster resides on the database server machine (**dbsrv1**) where the line-of-business database resides. The application is deployed to SilverMaster.

When this configuration makes sense This type of configuration is suitable when:

- The number of users is relatively low (under 50).
- The amount of data returned to the clients is small (such as a standard departmental application).
- Failover capabilities are not required. In some cases it may be acceptable to take the server down for infrequent administrative tasks such as a hardware upgrade or tape backup. So a clustered server arrangement is not a requirement.
- All users are authenticated against a single, existing security model. A department or site may have a preexisting server listing of users and groups (for example, a Windows domain), so this directory can be leveraged and used by the application server.

Benefits of this configuration This configuration has several benefits:

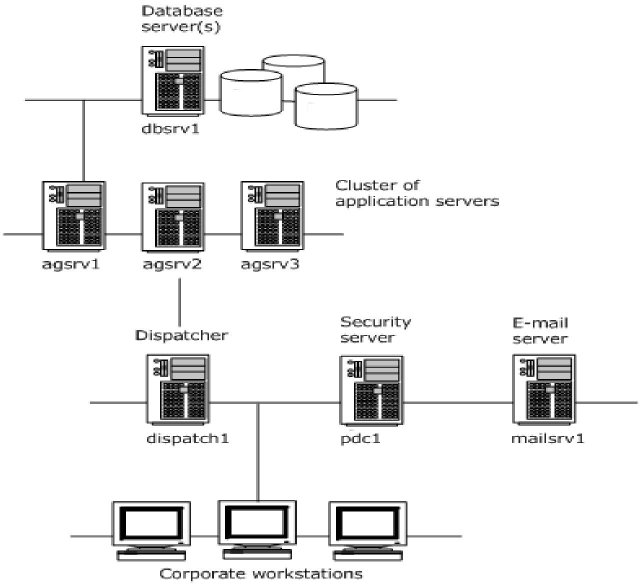
Benefit	Description
Simple administration	Administration of a single application server machine is easier than maintaining a group of computers hosting a cluster of servers.
Simple network topology	Because the number of users is small and the application complexity is low, there is no additional network configuration to make beyond ensuring proper TCP/IP connectivity.

Limitations of this configuration This configuration may be suitable for small companies or departmental applications, but there are some limitations to this approach:

Area	Limitation
Load balancing and failover	This solution offers no provision for maintaining application availability in the event of server downtime. A hardware failure on **agsrv1**, for example, would mean that no users could access the application until the problem was resolved.
Internet use	While suitable for smaller, intranet applications, this scenario provides no security mechanism for external use by Internet users. No firewall is provided to protect unauthorized access to unsecured LAN resources. And the Intranet security server (**pdc1**) is not used to authenticate external Internet or extranet users.

Intranet cluster configuration

In order to provide basic load balancing and failover capabilities, the application server provides a Dispatcher, Load Manager, and Cache Manager. The figure below shows a typical network diagram of several application servers (**agsrv1**, **agsrv2**, and **agsrv3**) in a cluster with traffic directed by the application server's software Dispatcher (**dispatch1**). The Cache Manager and Load Manager can reside on just about any machine in the network, though it is preferable to have them on the same physical subnet as the cluster of application servers.



In this scenario, a browser on one of the corporate workstations would access the application by connecting to the application server's software Dispatcher (**dispatch1**) using a browser or a Java application running in the SilverJ2EEClient container. Depending on the load plan, the Dispatcher would reply with an HTTP redirection to one of the available servers in the cluster.

To establish a connection, the client needs to resolve the TCP/IP host name of the target server using standard means. On Windows workstations, for example, the client would request the TCP/IP address of the target server from the WINS (Windows Internet Naming Service) or DNS (Domain Naming Service) server, or perform an NBT (NetBIOS over TCP/IP) broadcast to resolve the name and address. When they are resolved, the client would access the server directly. No subsequent trips to the Dispatcher would be made.

HTML application example A corporate user opens a browser to http://dispatch1/Accounting/default.html in order to log in to the company's accounting HTML application. The software Dispatcher (**dispatch1**) returns an HTTP redirect signal back to the client, which in turn establishes a connection directly to http://agsrv2/Accounting/default.html. Notice that not only was the browser redirected to the application server (**agsrv2**); the full URL address information (database name, Accounting, and page name) was also passed along.

The next user to access the application would be directed in round-robin fashion to the next available server according to the load plan: http://dispatch1/Accounting/default.html would be redirected to http://agsrv3/Accounting/default.html.

Benefits of this configuration This configuration has several benefits:

Benefit	Description
Server redundancy	In this load-balanced scenario, administrators are free to take down one or even two of the application servers for maintenance, because the other servers would be available for incoming requests provided that the remaining servers could accommodate the load.
Ease of administration	Setting up the cluster is extremely easy: initial configuration is wizard-based, and every aspect of server administration is done using the SMC. There’s no additional hardware or software (such as a third-party dispatcher or firewall) required to install and maintain this configuration.
Load balancing	This configuration is flexible: as the number of users grows, the number of servers can expand to accommodate them. The distribution of load across servers means that no one user can cause the server to be a bottleneck for other users in the organization.

Limitations of this configuration This configuration has some limitations:

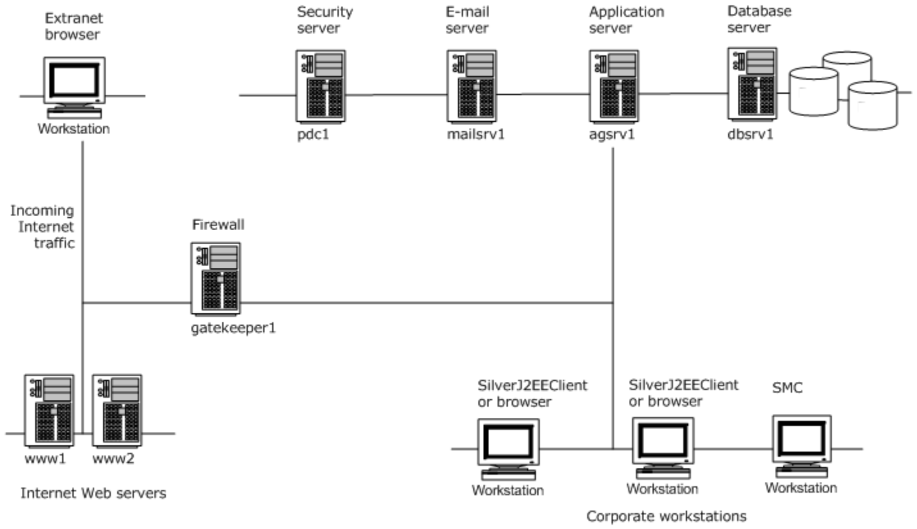
- No firewall or additional security mechanism is provided to protect unauthorized access to unsecured LAN resources.
- While this configuration could be used for Internet applications, there is no provision for DNS masking, such as the kind needed with a more advanced dispatcher. As such, all application servers and Dispatchers would need to be DNS-registered on the Internet.

To learn more For more information about clusters and load balancing, see Chapter 12, “Administering a Cluster”.

Simple Internet configuration

The figure below shows how a single application server can be used to provide extranet Web application functionality for both internal users (running a Java application using SilverJ2EEClient) and external business partners (accessing the HTML application over the Internet).

In this scenario, the application server (**agsrv1**) provides Web application services in conjunction with existing static content served from the corporate Web site servers (**www1** and **www2**). The application server (**agsrv1**) is DNS-registered; so when an extranet user is linked from the Web site to the application logon page (hosted on the application server), the browser knows what route to take in order to connect to the application server. In this case, Internet clients must pass through the firewall (**gatekeeper1**) in order to gain access to the application server.



To facilitate this connection, the firewall (**gatekeeper1**) has been configured so that only HTTP traffic on TCP/IP port 80 can pass through to the application server. This way system administrators are assured that the application-sensitive data will not be intercepted by someone other than the end user—and that incoming traffic cannot access other corporate resources.

The user accesses the application from a link on the corporate Web site (**www1** and **www2**). A Web server integration (WSI) module has been installed and configured on both Web servers and offers redirection capabilities to the logon page on **agsrv1**. Once redirected, browsers will establish a connection to the application server.

This process can be summarized as follows:

1. The user accesses an application server URL from one of the Web servers outside the firewall.
2. The WSI module responds to the browser with an HTTP redirection to the application server.
3. The browser automatically requests the URL directly from the application server, through the firewall.
4. For user authentication, upon connecting through the firewall to the application server (**agsrv1**), the user is prompted to log on to the application. A listing of extranet users is maintained in the server's SilverMaster database. This database, like the database serving the e-commerce application, is maintained on **dbsrv1**. The user enters the logon credentials and is logged on and can commence using the application.

Internal to the company, corporate users interact with extranet users using a Java application (using the SilverJ2EEClient container). For administrative purposes, corporate IT uses the SMC on HTTP port 80.

Benefits of this configuration This configuration has several benefits:

Benefit	Description
Secure e-commerce application	HTTP traffic between extranet users and the application server can pass safely over the Internet through the firewall. Administrators can log all logon activity using the firewall.
Ease of administration	Configuring the application server for use with the existing network was a simple case of adding a policy to the firewall configuration (for example, allow HTTP traffic to pass to **agsrv1** on TCP/IP 80 and log all activity).

Limitation of this configuration This configuration has the following limitation:

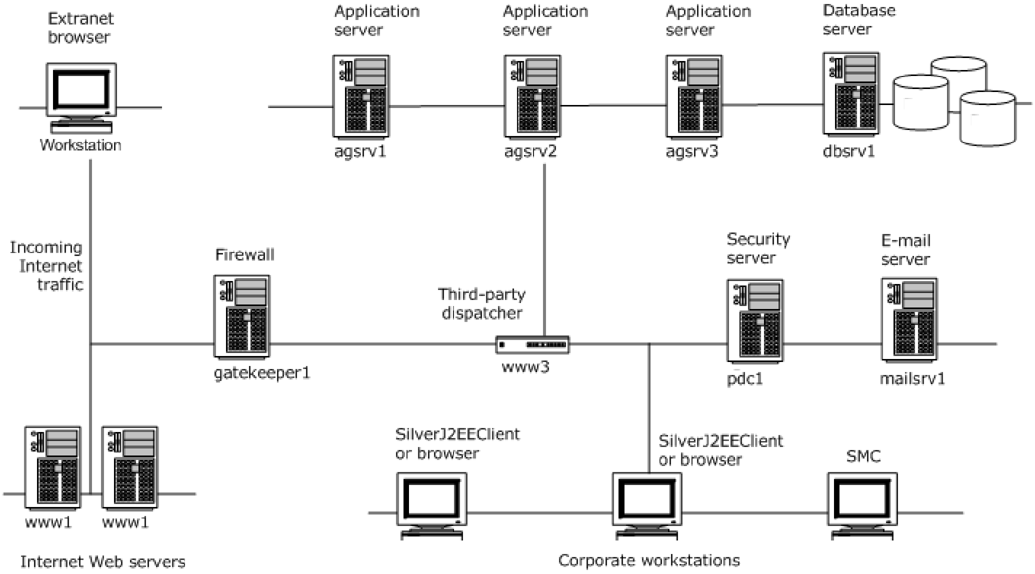
- **Load balancing and failover** Given that users inside the company and external business partners both use this application, the lack of load balancing and failover capabilities means that server downtime results in users not accessing the application.

To learn more For more information about WSI modules, see Chapter 8, “Using the Web Server Integration Modules”.

Internet cluster configuration

Larger-scale e-commerce applications usually require a very high degree of functionality, throughput, and availability. This requires an underlying system architecture that is more robust and complex than those previously shown.

The figure below shows an example of a large-scale Internet application served from a cluster of application servers. Internet users access the application using links from the two Web servers (**www1** and **www2**) located outside the firewall (**gatekeeper1**).



In order to implement transparent session-level failover and reduce overall DNS and firewall administration, the system administrators install a third-party hardware dispatcher that supports DNS masking, as opposed to using the application server’s software dispatcher. This way traffic to all application servers can be localized to a single TCP/IP address and host name on the Intranet (**www3**). In addition, with this type of device only one TCP/IP address and host name have to be DNS registered, as opposed to four machines when using the application server’s software dispatcher (**dispatch1**, **agsrv1**, **agsrv2**, and **agsrv3**).

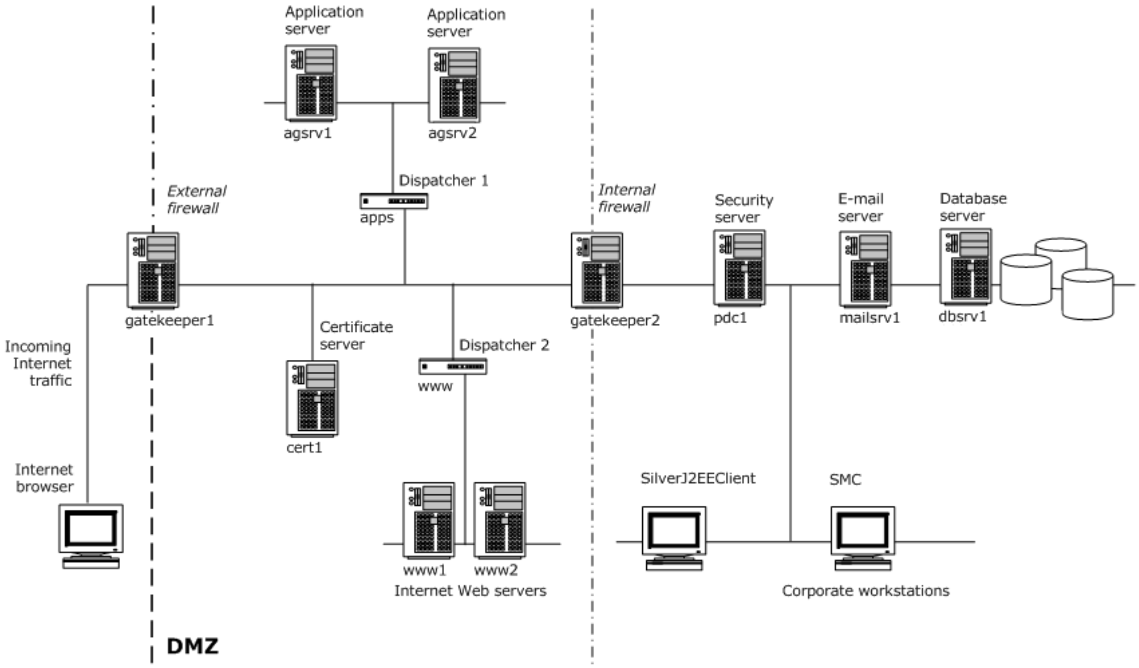
When any incoming requests are linked to the Web application itself (on **www3**), the browser establishes a connection through the firewall to the Web dispatcher. Based on its own load plan, the hardware dispatcher connects the browser to an available server in the cluster. Unlike the application server’s software dispatcher, the hardware dispatcher controls the flow of all HTTP traffic. In the event that the server goes down, the dispatcher can automatically route the browser session to a different server in the cluster. Since the dispatcher uses DNS masking, the failure is completely transparent to the end user.

To learn more For more information about clusters and load balancing, see Chapter 12, “Administering a Cluster”.

Demilitarized Zone (DMZ) Internet configuration

The complexity of Internet security and network infrastructure is often related to the size of a company. Larger companies with complex e-commerce Internet and extranet applications, for example, may have a two-tiered approach to firewall security.

The following figure shows such an example. All Internet traffic is routed through an Internet firewall (**gatekeeper1**). This firewall allows only Web traffic and Internet mail through to the Demilitarized Zone (DMZ), the area between the two firewalls. For security purposes, all Web and application servers reside in the DMZ.



It would have been possible to add a third network card to the firewall and have it protect Intranet traffic as well. However, for security reasons, this company decided to use separate devices.

DNS-masking hardware dispatchers (**www** and **apps**) are used to route traffic in a load-balanced fashion. It is also possible to use one device configured for multiple TCP/IP addresses and route traffic to both clusters. For redundancy purposes, however, two separate devices are used.

The Intranet firewall (**gatekeeper2**) allows e-mail traffic and database connections from the application servers (**agsrv1** and **agsrv2**) to pass through. This way the system administrators can be assured that only e-mail traffic and database calls from the secured DMZ (the application servers) can access corporate information.

External users can be authenticated by obtaining a browser certificate from the certificate server (**cert1**). The application servers can authenticate these users based on their certificates and encrypt the network traffic from the browser to the application server.

Benefits of this configuration This configuration is beneficial where there are the following requirements:

Benefit	Description
Security	System administrators have carefully designed this architecture to ensure that traffic from the outside world can only pass into the DMZ. For example, this multitier approach to security allows for tighter control over the origin of database access.
Availability	Server clusters are used for both static Web content and application services. Even the hardware dispatchers provide redundancy among themselves.
Session-level failover	The DNS-masking capabilities of the hardware dispatchers allow for this e-commerce application to run continuously, even in the event that a server fails (the user is automatically rerouted to another server).
High volume	The scalability of a multicluster server arrangement means that user load can be distributed among many servers. This is especially beneficial during peak periods of application usage.

This architecture is complex and more difficult to maintain than the average intranet site. However, you might want to set up this type of architecture to ensure the benefits listed above. Downtime often equates to loss of business. Maintaining a well-designed network infrastructure often pays for itself very quickly.

To learn more For more information about clusters and load balancing, see Chapter 12, “Administering a Cluster”.

Session management

The application server stores information about each client connection in a *session object*. A session is initiated when a client first connects to the server. Information in the session object includes information like user authentication. Applications can also store application-specific data in the session object. If a session containing a servlet or JSP page is idle for more than five minutes (the default), it is terminated. Use the SMC to change the session timeout value (by setting the **Timeout on server request** value).

📖 For more information on setting the **Timeout on server request** value, see “Setting performance parameters” on page 181.

The application server can use either cookies or URL rewriting to keep track of the state of multiple Web browser clients. Both cookies and URL rewriting use session IDs. All calls to the server within a browser session will operate under the same session ID. For secure data, authentication occurs once per active session for sessions requiring user authentication.

Authentication is the process through which the server and client verify one another’s identities. Authentication is described in “Enabling authentication” on page 159.

Cookies

The application server uses cookies to track sessions if the user’s browser supports them. A *cookie* is a piece of information sent by a Web server to a Web browser. The browser client stores the cookie and sends it back to the server whenever an additional request is made to the server. The application server uses the cookie as a session ID. When the server receives a request from a client that includes a cookie, it is able to use the information stored in the cookie to reconnect with the session.

IMPORTANT: The application server cookies are kept in memory and are never written to disk. There is no personal user or tracking information in the cookie.

If you or your users are concerned about the contents of cookies, you can set your browser not to accept cookies or to warn on cookies. For details about cookie, see your browser documentation .

URL rewriting

To support browsers that do not accept cookies, the application server rewrites URLs (with an appended **jsessionid** parameter) to correctly associate a request with a session. Application developers writing servlets or JSP pages need to understand how to use URL rewriting to support clients that do not accept cookies.

For information on URL rewriting for servlets and JSP pages, see "How session tracking works" next.

How session tracking works

The application server uses cookies if the user's browser supports them and uses URL rewriting if the browser does not. This determination happens at runtime for each user. The first time the application server receives a request, it sets a cookie—and also appends a **jsessionid** to the URL (because it does not yet know whether or not the client supports cookies).

When a client supports cookies, the application server will use them for session tracking (although it will rewrite the URL when it receives the first request). Once the client returns a cookie, the server will stop rewriting URLs for the client in this session.

NOTE: Cookies use the value **JSESSIONID** (all uppercase); URL rewriting uses **jsessionid** (all lowercase).

Administrator notes If the server determines that the client does not accept cookies, it uses the **jsessionid** in the URL for session tracking whenever the user clicks a link that is contained on a page.

The first time a client establishes a session, the URL **jsessionid** is appended to the URL and is visible to the client user. On subsequent interactions between server and client, the URL rewriting keeps track of the session ID, and the **jsessionid** is visible only when a user's mouse is held over a link on the page.

When a browser client does not accept cookies, servlets and JSP pages that use HTML links need to call one of two standard encode methods (described in "Developer notes" next) to rewrite URLs.

Developer notes The following encode methods enable the server to check for the existence of either a cookie or a **jsessionid**:

- HttpServletResponse.encodeURL()
- HttpServletResponse.encodeRedirectURL()

One of these encode methods is needed when servlets or JSP pages explicitly create or embed URLs in their responses.

The **jsessionid** in the URL is a path parameter, not a query parameter. *Query parameters* are usually at the end of the URL and separated by a **?**. *Path parameters* are at the end of a component of the URL and before any query parameters. In the following example:

```
http://server/db/foo;pparam=foo?qparam=bar&rparam=bar
```

pparam is a path parameter and **qparam** and **rparam** are both query parameters.

4 Data Source Configuration

This chapter describes how the Novell exteNd Application Server accesses and uses relational databases and Enterprise Information Systems (EIS) located on the data tier. It also describes how to set up access to these data sources. Topics include:

- About data sources
- Configuring deployment databases
- Configuring connection pools

About data sources

This section describes the different ways the application server uses databases and other components stored on the data tier:

Task	Description
System management	The application server uses a relational database, called **SilverMaster**, for overall system management. The server installation process creates and configures the SilverMaster database.
Deploying J2EE archives	The application server stores J2EE artifacts in a **deployment database**. A deployment database is a relational database added to the server. The artifacts are added to special system tables created and managed by the server. You use the SMC or SilverCmd to add the database to the server.
Accessing corporate data	The application server can access corporate data stored in an EIS or a relational database via a **connection pool**. You use the SMC to create, configure, and manage connection pools. You use the SMC or SilverCmd to add a connection pool to the server.

System management

The application server uses a master database catalog, called **SilverMaster**, for overall system management. The SilverMaster database is created and automatically added to the server during server installation. SilverMaster can be any **supported** database type that supports autoincrementing columns.

📖 For the complete list of supported database types, see the *Release Notes*.

SilverMaster functions

The SilverMaster database provides these system management functions:

- Stores information that spans all databases
- Maintains a catalog of databases managed by the server
- Tracks user and group authentication information
- Stores cluster information
- Stores deployed J2EE archives (optional)

The SilverMaster catalog contains internal tables that the application server uses for system management. These tables are reserved for application server use. For a list of these tables, see Appendix C, “System Tables and URLs”.

Default SilverMaster permissions After a default installation, users have Read access to the top level of the SilverMaster database and directories. This enables users to log in and access any existing deployment databases.

Users cannot add or remove any databases or access any deployed JARs until you grant them permission. If you have configured separate ports for different types of operations, you must use your administration port to update database configuration.

For information about troubleshooting the SilverMaster database, see “Using the SilverMasterInit program” on page 224.

Moving the SilverMaster database If you move your SilverMaster database from the host it was initially installed on, the application server needs to know the new connection location. The easiest way to update SilverMaster’s connection location information is to rerun the server’s installation program after moving the SilverMaster database.

NOTE: Some databases require you to update connection parameters (for example, by using ODBC, JDBC, or Oracle TNS).

In the server’s installation program, specify the moved SilverMaster database as you respond to the prompts. Be sure to choose the option **Install a new Server configuration file** from the screen that runs SilverMasterInit. It is not necessary to run SilverMasterInit.

If you run SilverMasterInit to initialize the SilverMaster properties, you will have to recreate any Silver Security users and groups, and manually add your deployment databases. As always, it is a good idea to test the connection to SilverMaster (using another application) before restarting the application server.

Deploying J2EE archives

J2EE applications are stored in archive files. The application server deploys J2EE archives to SilverMaster or a relational database that has been **added** to the application server. You add a database to the application server using the SMC or SilverCmd as described in “Configuring deployment databases” on page 46.

You can only add database types that are supported by the application server. For the complete list of supported databases, see the *Release Notes*.

What happens when you add a database to the server When you add a database to the server, an entry is made in SilverMaster so that the application server knows how to connect to and use the database—and system tables are added to the database too. The system tables are ordinary database tables, but they are reserved for application server use. The server stores the archives and any associated metadata in these system tables.

You can deploy an application to an already existing database (that contains corporate data) or you can add a database created as a target for application deployment.

NOTE: When adding a deployment database on a UNIX platform, you must add the location of the database to the AGCLASSPATH environment variable, then restart the server before you can add the database. For information about AGCLASSPATH, see "Setting the AGCLASSPATH variable" on page 96.

Adding databases in a restricted environment If your application server is running in a restricted production environment, you will need to authenticate yourself before adding (or deleting) a database. In a restricted environment, no users (except the server administrator) can add databases unless you grant them permission.

Accessing data The application server accesses corporate data via a connection pool, and not as an added database. If you deploy your J2EE archives to databases that also contain corporate that you want to access, you must create a connection pool for the database.

📖 For more information on adding connection pools, see "Configuring connection pools" on page 51.

Choosing a deployment database You can deploy your J2EE applications to any supported relational database or to the SilverMaster database. The following table describes some of the reasons for choosing SilverMaster or another deployment database:

Deployment database	Description
SilverMaster	You might choose to deploy all your J2EE archives to the SilverMaster database. The advantages include: ◆ The application's URL will not contain a database name ◆ You do not need to add and manage any other databases for the application server The disadvantages are: ◆ If you perform a SilverMasterInit the archives may be removed, and you will have to redeploy them (check out what SilverMasterInit does)
Non-SilverMaster databases	You might choose to add one or more relational databases to the server and deploy your J2EE archives to them. The advantages include: ◆ The application's URL will contain a database name ◆ If you perform a SilverMasterInit, the archives remain, and do not require a redeploy ◆ It is easier to configure the database connections to manage performance when the applications are in a separate database The disadvantages include: ◆ You have to manage more than one database

Accessing corporate data

J2EE applications (such as WARs, EARs, and EJB JARs) define the data sources they access as *resource references* in the deployment descriptor. When the archive is deployed to the server, the deployer uses deployment tools to map the resource reference to data sources available to the server.

As administrator you are responsible for making sure that the data source is available to the server and that the server has the appropriate permissions to the data source. You make a data source available to a J2EE application by creating a *connection pool*. You can create a connection pool using the SMC or SilverCmd as described in "Configuring connection pools" on page 51.

Database access

The application server can access relational databases through a native JDBC driver or through a JDBC-ODBC bridge driver.

Java Database Connectivity (JDBC)

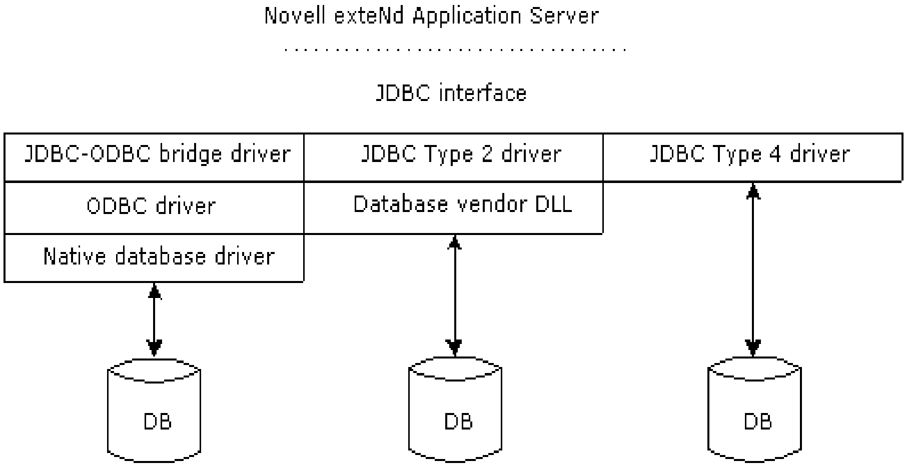
JDBC is a standard application program interface (API) for allowing Java applications such as the application server to enable SQL access to relational databases. The application makes JDBC calls to the JDBC driver, which translates the calls to the API of the underlying database. The Java runtime system supplies an ODBC bridge driver that allows JDBC to connect to supported databases through an ODBC driver.

JDBC access

There are four types of JDBC drivers:

JDBC driver	Description
Type 1: JDBC-ODBC bridge	For databases that support ODBC
Type 2: JDBC to a database vendor DLL	Supplied by the vendor or by third parties
Type 3: JDBC to middleware software to the database	Not recommended for use with the application server
Type 4: Pure Java to a network protocol	These are the best drivers to use with the application server, because they work directly with the network protocol

The following diagram shows the components of each supported JDBC driver type:



Adding JDBC driver JARs to the server classpath

To access a database via JDBC, the application server must be able to find the JAR files for the appropriate JDBC driver. That means those JARs must be added to the server's classpath. How this is done depends on the kind of database access you are setting up:

- For SilverMaster
- For any other database access

This section presents general guidelines for these setup tasks. To get the details for your DBMS, see the corresponding database configuration chapter in the application server's *Database Configuration Guide*.

For SilverMaster When you're setting up access to the SilverMaster database, adding JDBC driver JARs to the application server's classpath involves the following:

Platform	How JARs are added to server's classpath
NetWare	Before you can use any database drivers that are not the default for the server, you must use setenv to set the AGCLASSPATH environment variable to include the driver.
Windows	Before you install the application server, you must manually set up the system environment variable AGCLASSPATH to list the required JDBC driver JAR files.
UNIX	When you run the installation program for the application server, it automatically prompts for the location of your JDBC driver JAR files. Then it edits the .agprofile file (in the server's root directory) to set the AGCLASSPATH variable with that JAR information.

For any other database access When you're setting up any other database access (other than for SilverMaster) and a different database driver is involved, you must manually add those JDBC driver JAR files to the application server's classpath. This might occur if, for instance, you're accessing DB2 for SilverMaster but also need to establish connection pools for Oracle.

The typical way to do this is:

Operating system	Description
NetWare	Use setenv to set the AGCLASSPATH environment variable to include the JDBC driver files
UNIX	Edit the .agprofile file (in the server's root directory) to set the AGCLASSPATH variable with that JAR information
Windows	Edit the system environment variable AGCLASSPATH to add the required JARs

Testing your database connections

As you set up database access for the application server, it's recommended that you first test each connection using the tools provided by your database vendor. Knowing that those connections are valid can save time later if you need to troubleshoot access from the application server.

Configuring deployment databases

You can use a relational database as a deployment repository for your J2EE archives or as the application server's SilverMaster. The database you use can contain existing data or can be created only for deployment. This section describes how to make a deployment database available to the server and includes these topics:

- Preparing a deployment database
- Adding a deployment database to the server
- Removing a deployment database from the server
- Configuring a database

📖 For information on setting up a deployment database, see the chapter for your DBMS in the *Database Configuration Guide*.

Preparing a deployment database

The following table describes in general what you need to do so that you can use a database with the application server:

Task	Description
Set up a database user account for the application server	Use a DBMS utility for adding and modifying database account permissions (such as Sybase Central, Microsoft Enterprise Manager, Oracle Server Manager, Informix Control Center, and so on). The application server needs a database account to use when connecting to each database. The user account (such as Agsmith) must have CREATE TABLE, INSERT, UPDATE, and DELETE permissions. You should set up a separate account for your SilverMaster and one for each deployment database so you can easily tell how the database is being used. This strategy will help you more easily identify and troubleshoot performance problems. 📖 For information about other types of accounts, see "Administration accounts" on page 225.
Set up an ODBC data source for the database	ODBC control panel (Windows only; ODBC connections are currently not supported on UNIX).
Configure the DBMS client software on the application server machine	Use native database software (such as Oracle SQL-Net, Microsoft SQL Server client, Informix CLI, and so on).
Install the JDBC driver	Use native DBMS installer to install a JDBC driver on the application server machine (for example, jConnect).

📖 For information about how to set up a specific database type for use as SilverMaster or as a deployment database, see the appropriate configuration chapter in the *Database Configuration Guide*.

NOTE: You cannot name the database **SilverStream**.

Adding a deployment database to the server

Before you can deploy a J2EE archive to the server, you need to add the target deployment database (unless you are deploying to SilverMaster).

➤ **To add a database to the application server:**

1 Start the server.

2 Start the SMC.

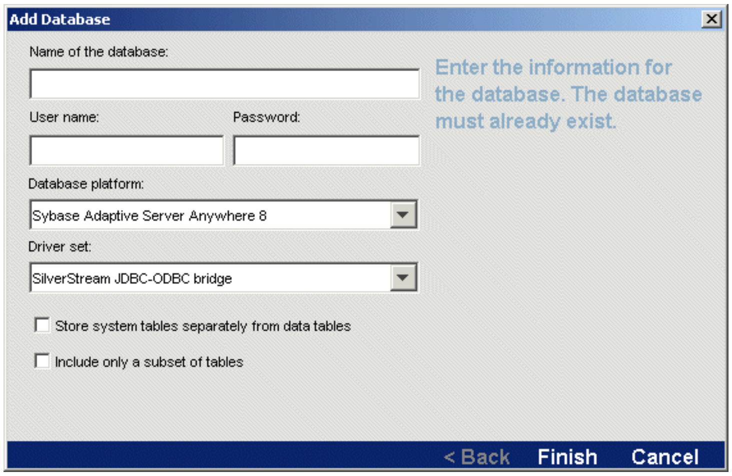
3 Select the server in the left pane of the SMC. If the server is not listed, add it to the SMC, as described in “Administering an application server remotely” on page 95.

4 Select the **Configuration** icon from the toolbar.

5 Select **Databases**.

6 Click **Add Database**.

You are prompted to enter information about the database:



Add Database

Name of the database:

User name:

Password:

Database platform: Sybase Adaptive Server Anywhere 8

Driver set: SilverStream JDBC-ODBC bridge

☐ Store system tables separately from data tables

☐ Include only a subset of tables

Enter the information for the database. The database must already exist.

7 Use the table below to enter the information for the database. If you need help, see the chapter for your DBMS in the *Database Configuration Guide*.

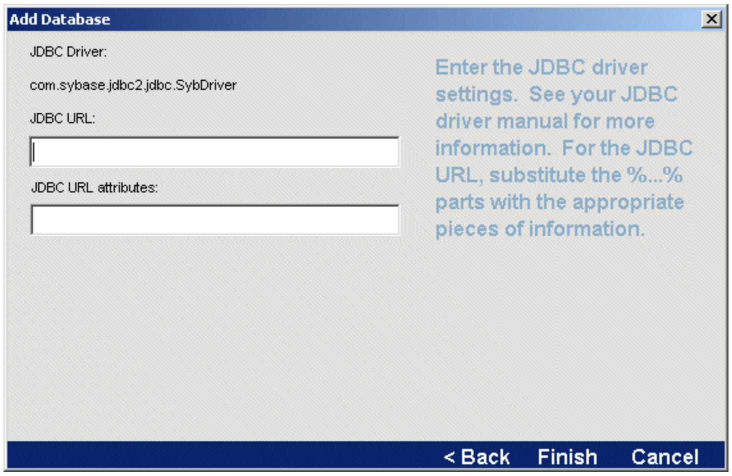
Field	What to specify
Name of the database	Enter the name of the database. For an ODBC database, the database name must be an already existing ODBC data source name.
User name and password	Enter a user name and password pair that the application server can use for a database user connection to your native database. These values cannot be null. This user name must already be known to the native database and have the appropriate read/write permissions. NOTE: Your administrator should have defined a unique user for each deployment database.
Database platform	Choose from the list of supported database platforms.
Driver set	Choose a driver set from the list. Driver sets are specific to the database platform you selected. The driver set recommended for your database type is displayed by default. Some driver sets require you to specify additional parameters. If you select one of these driver sets (a driver set whose name does not begin with **Novell exteNd**), see “Using another company’s driver set” on page 48.

Field	What to specify
Store system tables separately from data tables	If you plan to use a production database that other applications access, you may not want to add the application server's system tables to it. The system tables are used by the application server. This option allows you to store the application server's system tables in another database. If you check this option and click **Next**, a panel displays asking you to name the system table database. (This database must already exist.)
Include only a subset of tables	You may not want to use all the tables in the database you are adding. By selecting this option, you can specify a subset of tables to make available on the server. If you select this option and click **Next**, a panel displays with two list boxes. In the top box you can manually name each table you want to use. In the lower box you can specify patterns to indicate sets of tables; for example, you could specify **cust%** to use all tables starting with **cust**.

8 Click **Finish**.

The database is added to the server.

Using another company's driver set If you are not using an application server–supplied driver set, you must supply additional information:



JDBC Driver:

com.sybase.jdbc2.jdbc.SybDriver

JDBC URL:

JDBC URL attributes:

Enter the JDBC driver settings. See your JDBC driver manual for more information. For the JDBC URL, substitute the %...% parts with the appropriate pieces of information.

The following table describes each of the fields on this panel:

Field	What to specify
JDBC Driver	(Read-only) The fully qualified name of your JDBC driver class. For example: `com.sybase.jdbc.SybDriver` **NOTE:** Package names, like all Java names, are case sensitive.
JDBC URL	The URL string defined by the driver vendor to connect to your database. The string contains replaceable parameters surrounded by percent signs (%), such as %HOST%. For example: `jdbc:sybase:Tds:%HOST%:%PORT%/%DATABASE%` Substitute these parameters with values appropriate to your database.

Field	What to specify
JDBC URL attributes	Any additional URL attributes defined by the vendor that you can use to customize the driver connection. For example: `cache=100` Leave this field empty for DB2 databases.

📖 For more details, see your JDBC driver documentation.

What happens When you add a database, the server adds an entry in the SilverMaster database and also adds the application server's system tables to your database (unless you specified to keep system tables separate, in which case the system tables are added to the other database).

Adding a database from the command line You can also add a database to the server from the command line or from a batch file using the AddDatabase SilverCmd.

Deploying a J2EE application When you are ready to deploy your J2EE application, see the chapter on J2EE archive deployment in the *Facilities Guide*.

Moving an added database

If you have moved a database that you had added to the application server, remove the database from the server and then re-add it to the server.

- If you are using ODBC, you may also need to update the database's ODBC settings.
- If you are using a JDBC driver (such as jConnect), you will have to update the JDBC URL when adding the database back to the server.

Removing a deployment database from the server

If you don't need to maintain a connection between a database and the application server, you can remove the database from the server.

➤ **To remove a database from the server:**

1. Start the server.
2. Start the SMC.
3. Select the server in the left pane of the SMC. If the server is not listed, add it to the SMC, as described in "Administering an application server remotely" on page 95.
4. Select the **Configuration** icon from the toolbar.
5. Select **Databases**.
6. Select the database in the **Database settings** field.
7. Click **Remove Database**.
8. Click **OK**.

What happens When you remove a database connection from the server, the application server removes the entry from SilverMaster but leaves the database itself fully intact (including the application server's system tables).

Removing a database from the command line You can also remove a database from the server from the command line or from a batch file using the RemoveDatabase SilverCmd.

Configuring a database

You can use the SMC to configure databases that have been added to a server. For example, you can use the SMC to synchronize database information and delete idle connections. If you have configured separate ports for different types of operations, you must use your administration port to update database configuration.

➤ To configure a database:

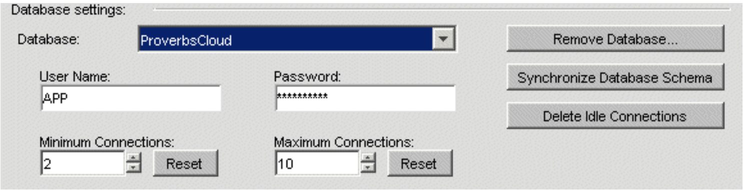
1 Start the SMC.

NOTE: If you have configured separate ports for different types of users and operations, you must specify your **administration** port to start the SMC.

2 Select the **Configuration** icon from the toolbar.

3 Select **Databases**.

4 Select the database name from the dropdown list:



Database: ProverbsCloud

User Name: APP

Password: **********

Minimum Connections: 2

Maximum Connections: 10

5 Enter information for the database as follows:

Field	Description
User Name and Password	A user name and password pair, which the application server can use for a database user connection to your database. The user name must already be known to the database and have the appropriate Read/Write permissions.
Minimum Connections	The minimum number of server connections for this database.
Maximum Connections	The maximum number of server connections for this database.
Remove database	A button that removes the selected database from the server. See "Removing a deployment database from the server" on page 49
Synchronize Database Schema	A button that lets you synchronize the application server's image of the database with any changes to the database structure. The application server keeps its own image of the database schema. When you modify the database's structure, click this button to update the server's image of it. 🕮 For more information about this option, see "Synchronizing the database schema" on page 51.
Delete Idle Connections	A button that releases database connections that are not currently being used. When the server needs more connections, the connection pool will automatically regrow as needed to the maximum number of connections defined. Deleting idle connections allows you to (at least temporarily) free up some database connections for use by other applications without having to restart the server. 🕮 For information on permanently changing the pool size, see "Setting the maximum and minimum number of database connections" on page 192.

Field	Description
System Database Properties	A button that displays only if the selected database is storing its application server system tables in another database (you specify this property when adding a database to a server). Click this button to see information about the database that stores the application server's system tables for the selected database. 📖 For more information about storing system tables separately, see AddDatabase in the SilverCmd reference chapter in the *Facilities Guide*.

Synchronizing the database schema You may need to synchronize the server metadata and the current database schema to ensure that tables, views, and key definitions cached on the server match the current database structure. This is not checked by default. Use the -dbcheck command-line option to force this check to occur.

NOTE: You can also use the **-noexitondbcheck** command-line option to see any errors while still starting the server. If you see any errors, you should synchronize the database. For more information, see "Database not synchronized" on page 223.

When it receives a request to synchronize the database, the server:

- Generates the current database schema
- Synchronizes the generated schema with the cached schema
- Sends a response back to the client if it was successful or sends an exception if it failed

Configuring connection pools

You make corporate data in relational database or EIS systems available to the application server via connection pools. This section describes how to create and maintain connection pools. It includes these sections:

- Preparing a connection pool
- Adding a JDBC connection pool
- Adding a Connector connection pool
- Removing a connection pool
- Maintaining a connection pool
- Connection pool considerations

Preparing a connection pool

Before you create a connection pool, make sure you've performed the administrative tasks outlined in the following table:

Data source type	Administrative tasks
Relational databases	Install a JDBC driver on the server Create a user ID and password for the application server to use Create a **JDBC connection pool**
EIS	Deploy a resource adapter archive (RAR) on the server Create a user ID and password for the application server to use Create a **Connector connection pool**

The application server uses connection pools to provide access to corporate data in relational databases (via JDBC) or one or more EIS (via a RAR deployed to the server).

Adding a JDBC connection pool

This section describes how to add JDBC connection pools using the SMC's Add JDBC Connection Pool Wizard. When you create a JDBC connection pool, you must have a JDBC driver installed on your system; the application server supports both JDBC 1.0 and JDBC 2.0 drivers.

This section provides the following topics:

- Panel sequence
- Starting the Add JDBC Connection Pool Wizard
- Panel reference

From the command line You can also add a JDBC connection from the command line. See "Adding a connection pool from the command line" on page 59.

Panel sequence

The wizard panels and the order in which they are presented vary depending on the type of JDBC driver you are using to access the database. This table shows how you step through the wizard based on the type of JDBC connection pool you want the wizard to build. You can click the links to get more information about the values you must supply for each panel.

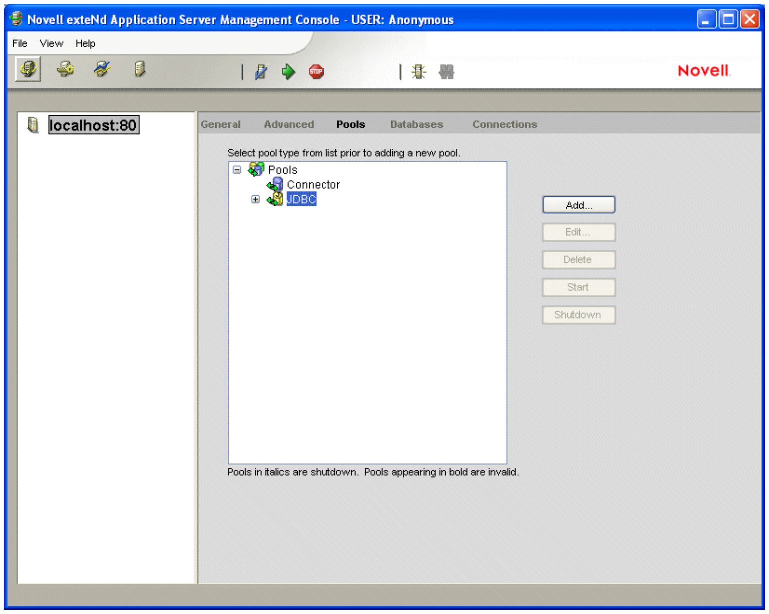
If you want to create a JDBC connection pool	You'll be responsible for
For JDBC drivers that are preconfigured for exteNd *	**1** Starting the Add JDBC Connection Pool Wizard **2** Specifying a preconfigured or user-specified driver **3** Specifying the pool name **4** Specifying the JDBC driver and URL **5** Specifying data source configuration properties **6** Specifying connection and timeout properties
For JDBC 1.0 user-defined drivers	**1** Starting the Add JDBC Connection Pool Wizard **2** Specifying a preconfigured or user-specified driver **3** Specifying the JDBC version **4** Specifying the pool name **5** Specifying the JDBC driver and URL **6** Specifying data source configuration properties **7** Specifying connection and timeout properties
For JDBC 2.0 user-defined drivers	**1** Starting the Add JDBC Connection Pool Wizard **2** Specifying a preconfigured or user-specified driver **3** Specifying the JDBC version **4** Specifying the data source information **5** Specifying the pool name **6** Specifying data source configuration properties **7** Specifying connection and timeout properties

* A preconfigured driver is a driver for which the application server provides a higher level of service. For preconfigured drivers, the application server knows how to handle error codes returned by the driver and can also work around bugs in the driver.

Starting the Add JDBC Connection Pool Wizard

➤ To start the Add JDBC Connection Pool Wizard:

1. Start the server.
2. Start the SMC.
3. Select the server in the left pane of the SMC. If the server is not listed, add it to the SMC, as described in “Administering an application server remotely” on page 95.
4. Select the **Configuration** icon from the toolbar.
5. Select **Pools**.
6. Choose **JDBC** and click **Add**:



Select pool type from list prior to adding a new pool.

Pools in italics are shutdown. Pools appearing in bold are invalid.

For more information about how to continue, see “Panel sequence” on page 52.

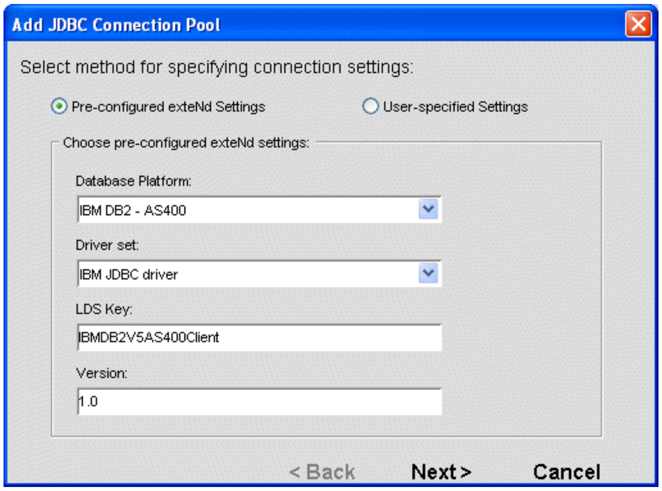
Panel reference

This section contains reference information for these tasks:

- Specifying a preconfigured or user-specified driver
- Specifying the pool name
- Specifying the JDBC driver and URL
- Specifying data source configuration properties
- Specifying connection and timeout properties
- Specifying the JDBC version
- Specifying the data source information

Specifying a preconfigured or user-specified driver

This panel is used to specify whether you are using a preconfigured or user-defined driver.



Select method for specifying connection settings:

● Pre-configured exteNd Settings ○ User-specified Settings

Choose pre-configured exteNd settings:

Database Platform: IBM DB2 - AS400

Driver set: IBM JDBC driver

LDS Key: IBMDB2V5AS400Client

Version: 1.0

1 Complete the panel as follows:

Field	What to specify
Pre-configured exteNd settings	Choose this option if the JDBC driver you are using is listed in the Database Platform and Driver set dropdowns.
User-specified Driver	Choose this option if you want to create a connection pool for a JDBC that is not preconfigured.

2 If you chose User-specified Driver, click **Next**.

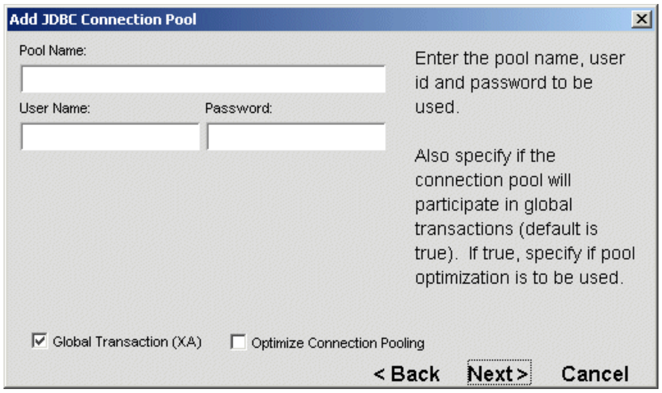
3 If you chose Pre-configured exteNd settings, complete the remaining fields as follows:

Field	What to specify
Database Platform	Choose from the list of supported database platforms.
Driver set	Choose the driver set from the list (a driver set is a JDBC driver, sometimes in combination with application server–specific files). The listed driver sets are specific to the database platform you selected. The driver set recommended for your database type is displayed by default.
LDS Key	Read-only field that displays the actual key associated with the related database platform and driver set.
Version	Read-only field that displays the supported JDBC version of the selected driver.

4 Click **Next**.

Specifying the pool name

This panel is used to specify the name of the pool and the username/password combination the server will use to connect to the target database:



Enter the pool name, user id and password to be used.

Also specify if the connection pool will participate in global transactions (default is true). If true, specify if pool optimization is to be used.

☑ Global Transaction (XA) ☐ Optimize Connection Pooling

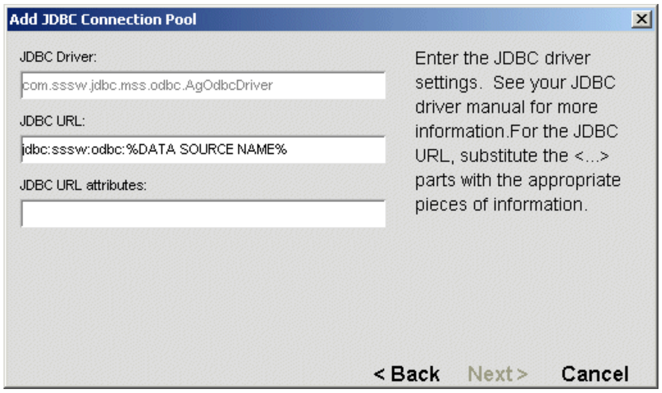
1 Complete the panel as follows:

Field	What to specify
Pool Name	Enter the name for the connection pool. This name must be unique on the server. It is the name that J2EE resource references will use to connect to the database. Limited to 32 characters.
User Name and Password	Enter a user name/password pair that the server can use for a user connection to the native database. These values cannot be null. This user name must already be known to the native database and have the appropriate read/write permissions.
Global Transaction (XA)	If this is checked (the default), connections returned by this pool can be enlisted in global transactions. If this is not checked, connections returned by this pool cannot be enlisted in global transactions even if one is active at the time of the request.
Optimize Connection Pooling	When checked, this option allows for more efficient handling of connections shared in a transaction. This option is only applicable to pools representing JDBC drivers that support the XA standard and when your JDBC driver is able to gracefully handle active delistment of resources from reenlistment into a transaction.

2 Click **Next**.

Specifying the JDBC driver and URL

This panel lets you specify information about the JDBC driver you are supplying:



Add JDBC Connection Pool

JDBC Driver: com.sssw.jdbc.mss.odbc.AgOdbcDriver

JDBC URL: jdbc:sssw:odbc:%DATA SOURCE NAME%

JDBC URL attributes:

Enter the JDBC driver settings. See your JDBC driver manual for more information.For the JDBC URL, substitute the <...> parts with the appropriate pieces of information.

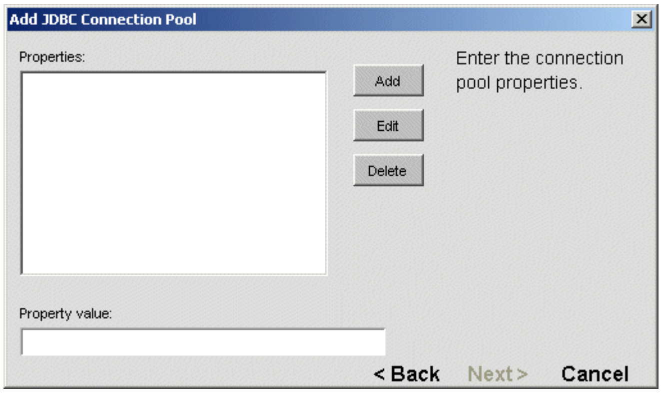
1 Complete the panel as follows:

Field	What to specify
JDBC Driver	(Read-only) Displays the fully qualified name of your JDBC driver class. For example: `com.sybase.jdbc.SybDriver`
JDBC URL	The URL string defined by the driver vendor to connect to your database. The string contains replaceable parameters surrounded by percent signs (%), such as %HOST%. For example: `jdbc:sybase:Tds:%HOST%:%PORT%/%DATABASE%` Substitute these parameters with values appropriate to your database.
JDBC URL attributes	Any additional URL attributes defined by the vendor that you can use to customize the driver connection. For example: `cache=100`

2 Click **Next**.

Specifying data source configuration properties

This panel lets you provide any additional properties for the connection pool your JDBC driver can support:



Add JDBC Connection Pool

Properties:

Property value:

Enter the connection pool properties.

1 To supply the properties, choose **Add** and then use the following table to complete the panel:

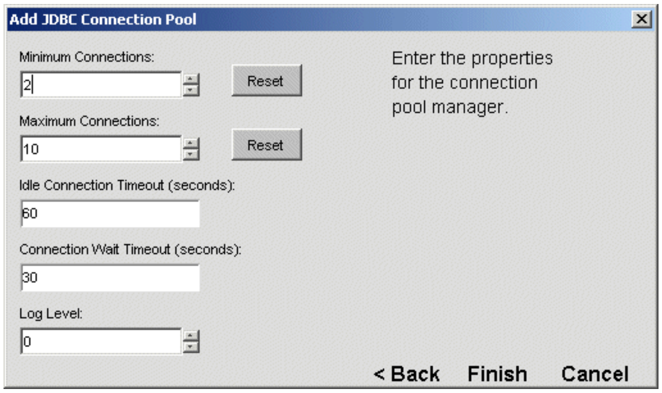
Field	What to specify
Property Name	Name of the ManagedConnectionFactory property
Property value	Value of the ManagedConnectionFactory property

NOTE: These values are determined by the driver you are using. For more information on these properties, see the vendor's documentation.

2 Click **Next**.

Specifying connection and timeout properties

This panel lets you specify connection and timeout values for the connection pool:



Add JDBC Connection Pool

Minimum Connections: 2

Maximum Connections: 10

Idle Connection Timeout (seconds): 60

Connection Wait Timeout (seconds): 30

Log Level: 0

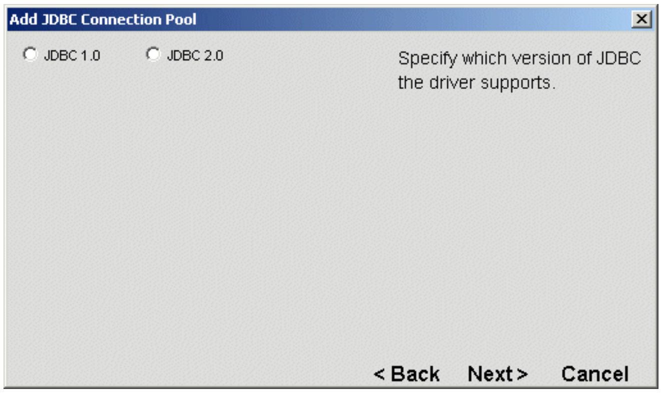
Enter the properties for the connection pool manager.

1 Complete the panel as follows:

Field	What to specify
Minimum Connections	The minimum number of connections. The pool manager will attempt to maintain this minimum number of connections. (This is a soft limit.)
Maximum Connections	The maximum number of connections allowed by the pool. The default is 10. To create a pool with no maximum, use -1.
Idle Connection Timeout	The amount of time (in seconds) that a connection (in the connection pool) is idle before the application server closes it. The default is 60 seconds. When set to -1, idle timeout is disabled and no idle connections are ever closed.
Connection Wait Timeout	The amount of time (in seconds) that an application component will wait for a connection from the pool. The default is 30 seconds. When it is set to -1, clients are forced to wait until a connection becomes available.
Log Level	The levels are: 0—Logging is turned off 1—Logs basic connection pool operations 2—Level 1 with more detailed operations and error messages 3—Level 2 with exception stack traces and trace output produced by JDBC driver or Connector resource adapter Messages are written to the server console.

Specifying the JDBC version

This panel lets you specify the version for your JDBC driver:



Specify which version of JDBC the driver supports.

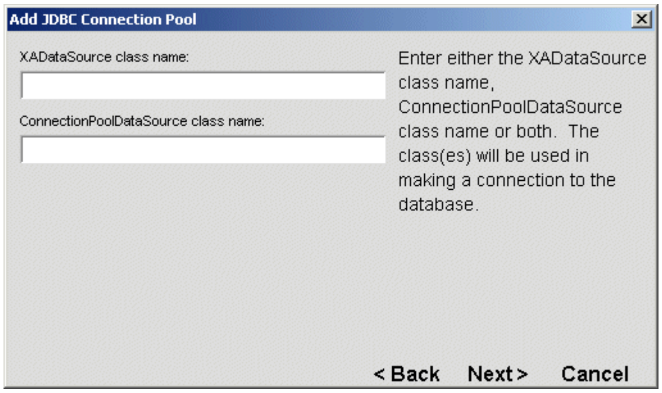
1 Complete the panel as follows:

Field	What to specify
JDBC 1.0	Choose this option if your JDBC driver supports JDBC 1.0
JDBC 2.0	Choose this option if your JDBC driver supports JDBC 2.0

2 Click **Next**.

Specifying the data source information

This panel lets you specify the datasource class name and or connection pool class name for JDBC 2.0 drivers:



Enter either the XADataSource class name, ConnectionPoolDataSource class name or both. The class(es) will be used in making a connection to the database.

1 Use the following table to complete the panel (you must enter a value for at least one of the fields):

Field	What to specify
XADataSource class name	Specify the fully qualified name of the XA DataSource class
ConnectionPoolDataSource class name	Specify the fully qualified Connection Pool DataSource class name

Even if you specify both the ConnectionPoolDataSource and XADataSource class names, only one is used. Which one that is depends on the overall configuration properties you specified in the panel described in "Specifying data source configuration properties" on page 56. The configuration properties are then applied to the instance of the data source class.

2 Click **Next**.

What happens

When you add a connection pool, the application server creates the connection to the database with the user name you specified and preallocates the minimum number of connections you specified.

Adding a connection pool from the command line

You can also add a connection pool from the command line or from a batch file using the AddCP SilverCmd.

Adding a Connector connection pool

When you create a Connector connection pool, you must have a RAR deployed and enabled on the server. For more information on deploying a RAR, see J2EE Archive Deployment in the *Facilities Guide*.

This section describes how to add connection pools using the SMC's Add Connector Connection Pool Wizard.

➢ **To add a Connector connection pool:**

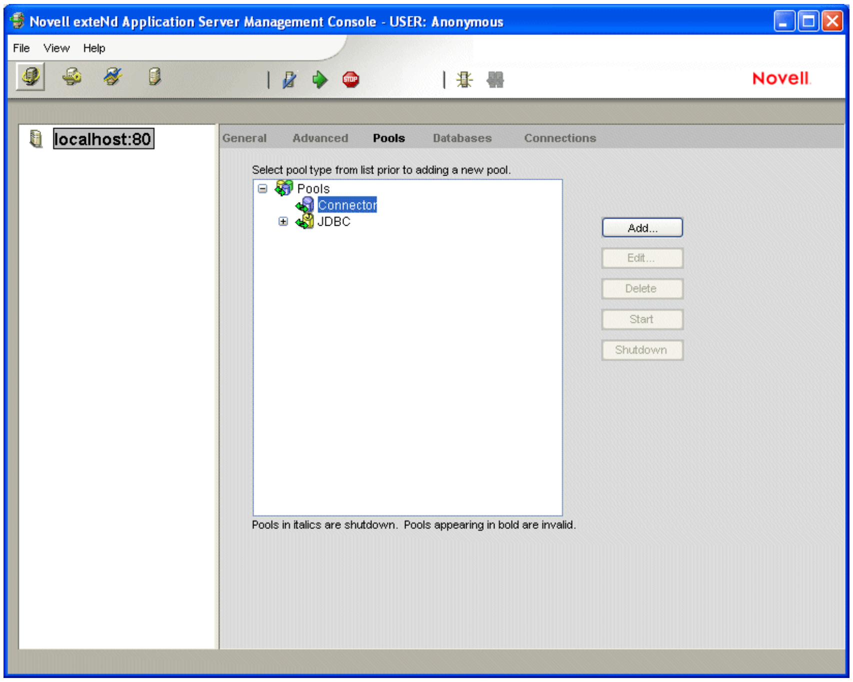
1 Start the server.

2 Start the SMC.

3 Select the server in the left pane of the SMC. If the server is not listed, add it to the SMC, as described in "Administering an application server remotely" on page 95.

4 Select the **Configuration** icon from the toolbar.

5 Select **Pools**:

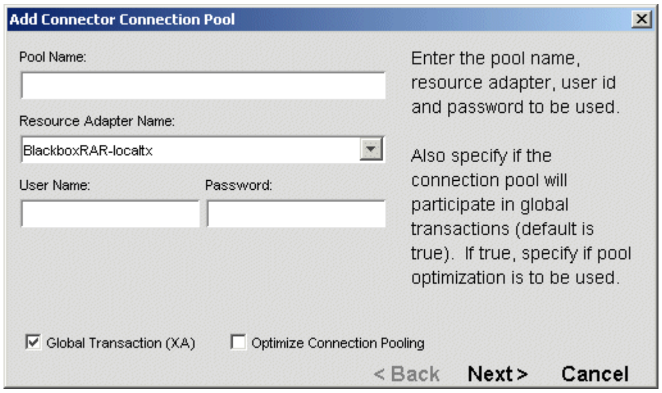


Novell exteNd Application Server Management Console - USER: Anonymous
File View Help
Novell
localhost:80
General Advanced Pools Databases Connections
Pools
Connector
JDBC
Add...
Edit...
Delete
Start
Shutdown

Select pool type from list prior to adding a new pool.

Pools in italics are shutdown. Pools appearing in bold are invalid.

6 Select **Connector** and click **Add**. You are prompted to specify the name of the pool and the username/password combination the server will use to connect to the target database:



Add Connector Connection Pool

Pool Name:

Resource Adapter Name: BlackboxRAR-localtx

User Name:

Password:

Enter the pool name, resource adapter, user id and password to be used.

Also specify if the connection pool will participate in global transactions (default is true). If true, specify if pool optimization is to be used.

☑ Global Transaction (XA)

□ Optimize Connection Pooling

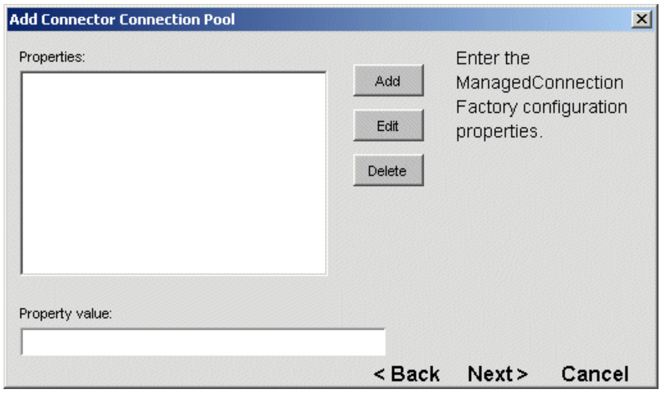
7 Provide the connection pool information as follows:

Field	What to specify
Pool Name	Enter the name for the connection pool. This name must be unique on the server. This is the name J2EE applications that contain resource references will use to connect to the data source. Limited to 32 characters.
Resource Adapter Name	Enter the name under which the RAR was deployed. If you do not know the name of the deployed RAR, see the Resource Adapters panel of the SMC (available via the Deployment icon on the toolbar).
User Name and Password	Enter a user name/password pair that the application server can use for a user connection to the EIS. These values cannot be null. This user name must already be known to the EIS and have the appropriate read/write permissions.

Field	What to specify
Global Transaction (XA)	If this is checked (the default), connections returned by this pool can be enlisted in global transactions. If this is not checked, connections returned by this pool cannot be enlisted in global transactions even if one is active at the time of the request. If you check the XA option and the driver you want to use is identified by LDS key, then the server will make a decision about which type of the connection factory to use that will be best suited for supporting XA. For 2.0 drivers that provide an implementation of XADataSource, XADataSource is the best option. An XADataSource is a JavaBean and requires a set of properties instead of a single URL. This is why the SMC does not enable the JDBC URL field and instead prompts you to enter the set of configuration properties for the XADataSource.
Optimize Connection Pooling	When checked, this option allows for more efficient handling of connections shared in a transaction. This option is applicable only to pools representing Connectors that support: ◆ The XA standard ◆ The graceful delistment of resources from reenlistment into a transaction

8 Click **Next**.

You are prompted to enter vendor-specific properties the pool should support:



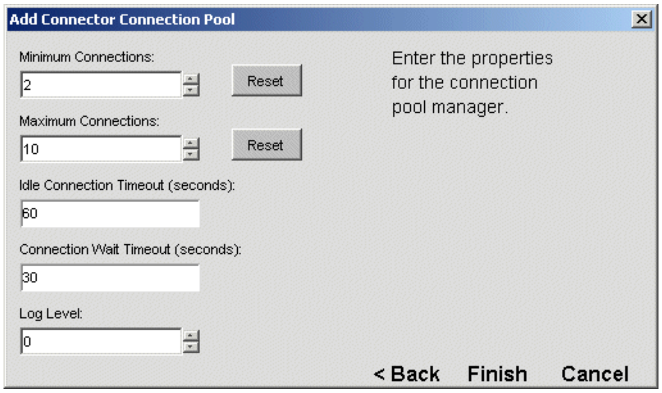
Enter the ManagedConnection Factory configuration properties.

9 To supply the properties, click **Add**.

10 Complete the panel as follows:

Field	What to specify
Property Name	Name of the ManagedConnectionFactory property
Property Value	Value of the ManagedConnectionFactory property

You are prompted to enter information about the pool connections and connection timeout values:



Add Connector Connection Pool

Minimum Connections: 2

Maximum Connections: 10

Idle Connection Timeout (seconds): 60

Connection Wait Timeout (seconds): 30

Log Level: 0

Enter the properties for the connection pool manager.

11 Complete the panel as follows:

Field	What to specify
Minimum Connections	The minimum number of connections. The pool manager will attempt to maintain this minimum number of transactions. (This is a soft limit.)
Maximum Connections	The maximum number of connections allowed by the pool. The default is 10. Use –1 to create a pool with no maximum.
Idle Connection Timeout	The amount of time (in seconds) that a connection (in the connection pool) is idle before the application server closes it. The default is 60 seconds. When this field is set to –1, idle timeout is disabled and no idle connections are ever closed.
Connection Wait Timeout	The amount of time (in seconds) that an application component will wait for a connection from the pool. The default is 30 seconds. When set to –1, clients are forced to wait until a connection becomes available.
Log Level	The levels are: 0—Logging is turned off 1—Logs basic connection pool operations 2—Level 1 with more detailed operations and error messages 3—Level 2 with exception stack traces and trace output produced by JDBC driver or Connector resource adapter Messages are written to the server console.

12 Click **Finish**.

What happens When you add a connection pool, the application server creates the connection to the EIS with the user name you specified and preallocates the minimum number of connections that you specified.

Adding a connection pool does not require you to restart the server—unless you remove a connection pool and then add a pool (of the same type) with the same name. If the original connection pool (the one that was removed) was used by a running application, it is possible that one of the active components, such as an EJB object in the pool, has cached a reference to a java.sql.DataSource object. This reference might refer to the invalid connection pool. Restarting the server will clear the cached references.

Adding a connection pool from the command line You can also add a connection pool from the command line or from a batch file using the AddCP SilverCmd.

Removing a connection pool

If you don't need to maintain a connection between a database or EIS and the application server, you can remove the connection pool from the server.

➤ **To remove a connection pool from the server:**

1. Start the server.
2. Start the SMC.
3. Select the server in the left pane of the SMC. If the server is not listed, add it to the SMC, as described in "Administering an application server remotely" on page 95.
4. Select the **Configuration** icon from the toolbar.
5. Select **Pools**.
6. Select the connection pool from the list.
7. Click **Delete**.
8. Click **OK**.

Removing a connection pool from the command line You can also remove a database from the server from the command line or from a batch file using the RemoveCP SilverCmd.

Maintaining a connection pool

You can edit a subset of connection pool properties, shut down a pool, and restart a pool.

➤ **To edit connection pool properties:**

1. Start server.
2. Start the SMC.
3. Select the server in the left pane of the SMC. If the server is not listed, add it to the SMC, as described in "Administering an application server remotely" on page 95.
4. Select the **Configuration** icon from the toolbar.
5. Select **Pools**.
6. Select the connection pool from the list.
7. Click **Edit**.

   The Edit Connection Pool Wizard displays. You can change different properties of the connection pool depending on its type. For information about the JDBC connection pool panels, see "Panel reference" on page 53; for information about the connector connection pool panels, see "Adding a Connector connection pool" on page 59.
8. Click **Next** to navigate the Edit Connection Pool Wizard to modify the properties.
9. Click **Finish** to complete the wizard. You do not need to restart the connection pool or the server for the changes to take effect.

Shutting down a connection pool

When you shut down a connection pool, it is not available to service client connection requests.

You may want to shut down a connection pool when the underlying database or EIS is temporarily brought offline—because it guarantees that the pool will not service user connection requests.

When you shut down a connection pool, all database connections are closed and all resources associated with the connections are freed. Use **Restart** (described below) to make the connection pool available again. Restarting the server will **not** restart a shut down connection pool.

➢ **To shut down a pool:**

1. Start the server.
2. Start the SMC.
3. Select the server in the left pane of the SMC. If the server is not listed, add it to the SMC, as described in "Administering an application server remotely" on page 95.
4. Select the **Configuration** icon from the toolbar.
5. Select **Pools**.
6. Select the connection pool from the list.
7. Click **Shutdown**.

   You are asked to confirm the action.
8. Click **Yes** to shut down the pool.

   The names of pools that are shut down appear in italic.

Recognizing an invalid connection pool If a pool name displays in bold, that means it is invalid. A pool might be invalid if the connection pool failed to start at server initialization time. This may happen when the database is down or if some network problem occurs and connections cannot be created. To remove any invalid pools, see "Removing a connection pool" next.

Restarting a connection pool

You can restart a connection pool that was stopped by the **Shutdown** button of the SMC.

The pool is restarted using the configuration properties (such as minimum/maximum connections, timeouts, and so on) used by the pool before it was shut down.

➢ **To restart a pool:**

1. Start your server.
2. Start the SMC.
3. Select the server in the left pane of the SMC. If the server is not listed, add it to the SMC, as described in "Administering an application server remotely" on page 95.
4. Select the **Configuration** icon from the toolbar.
5. Select **Pools**.
6. Select the connection pool from the list.
7. Click **Restart**.
8. Click **Yes**.

Connection pool considerations

This section describes other considerations that apply in creating and managing connection pools and includes these topics:

- Specifying how the application server reclaims connections
- Using JDBC drivers and Connectors that support JTA/XA
- Using connection pools configured to enlist global transactions
- Using container-managed sign-on

Specifying how the application server reclaims connections

The application server reclaims connections used by application components in different ways depending on the type of component (stateful or stateless):

- For **stateless** components, like servlets and stateless session beans, the application server reclaims connections after the method invocation
- For **stateful** components, like stateful session beans, the application server will not reclaim a connection as long as the bean is actively used by a client

You can configure how the application server reclaims connections opened by stateless components using the http-server.com.sssw.srv.invctx.releaseRes property in the httpd.props file. The property has these values:

Value	Description
True	(The default.) The application server returns connections to the connection pool automatically after each server invocation of a servlet or stateless session bean. This ensures that any connections not properly closed by an application are not held by the application—possibly using up the available connections in the connection pool.
False	Connections are **not** returned to the connection pool automatically after each invocation of a servlet or stateless session bean.

Using JDBC drivers and Connectors that support JTA/XA

You should use JDBC drivers and resource adapters that provide JTA/XA support—especially with applications and components that perform transactional processing.

When you create connection pools for JDBC drivers that do **not** support the JTA/XA protocol or Connector resource adapters that support **only** local transactions, the application server enables connections created by these pools to participate in global transactions—but only one such connection can participate in a transaction. By default, the application server attempts to share the connections obtained in the scope of the transaction (unless specified by the deployment descriptor). This means that:

- If the component attempts to obtain more than one unshared connection from the pool, the pool manager will throw an exception when the second connection request is made
- If one or more components involved in the same transaction attempt to obtain two connections from the pool using different security credentials (such as user names and passwords), the connection pool throws an exception when the second request is made

Work performed using these types of connections is not recoverable.

Using connection pools configured to enlist global transactions

JDBC and Connector-based connection pools by default are configured to enlist connections in global transactions. You can create a connection pool whose connections are **not** enlisted in a global transaction when the transaction:

- Is active at the time of the connection request
- Was started via the UserTransaction interface

Do not configure J2EE applications or standalone components to use connection pools configured in this manner. They will violate the application's or component's transactional semantics.

Using container-managed sign-on

Use **container** managed sign-on when possible. Applications that use **component** managed sign-on are less portable and less efficient.

II Administering the Server

This part describes the tasks you will perform to administer the Novell exteNd Application Server

- Chapter 5, “Running the Server”
- Chapter 6, “Setting Up Users and Groups”
- Chapter 7, “Maintaining the Server”
- Chapter 8, “Using the Web Server Integration Modules”
- Chapter 9, “Setting Up Security”
- Chapter 10, “Using Security”
- Chapter 11, “Tuning the Server”
- Chapter 12, “Administering a Cluster”
- Chapter 13, “Using the Server Administration API”
- Chapter 14, “Troubleshooting”

5 Running the Server

This chapter describes how to run the Novell exteNd Application Server. It contains sections on:

- Starting the application server
- Shutting down the application server
- Restarting the application server
- Maintaining processes running as services
- Setting up separate ports
- Specifying general server properties
- Using server logging
- Specifying ORB settings
- Running multiple servers on one host
- Specifying character set encoding
- Running the JMS server

Starting the application server

This section provides platform-specific information for manually starting the application server.

NOTE: You can also run the application server in the background as a service in Windows or as a daemon in UNIX. For more information, see *Installing Novell exteNd*.

This section contains the following topics:

- Starting the application server
- Using startup options
- Specifying the JVM to use
- Starting the server on a specific IP address or hostname

Starting the application server

➤ **To start the application server in Windows:**

- Do one of the following:
  - From **Start>Programs>Novell exteNd *n.n*>AppServer**, select **Application Server**.
  - Run the application server from a DOS command line by issuing the **SilverServer** command with one or more startup options.

📖 For more information, see "Using startup options" below and "Specifying the JVM to use" on page 72.

➢ **To start the application server on UNIX or Linux:**

1. From the command line, change to the server's **\bin** directory.
2. Type **./SilverServer** and any startup options. To print a list of available options, enter:

```
./SilverServer -?
```

📖 For more information, see "Using startup options" below and "Specifying the JVM to use" on page 72.

➢ **To start the application server on NetWare:**

1. From the system console, type **silverserver** and any startup options. To print a list of available options, enter:

```
silverserver -?
```

OR

```
silverserver -help
```

Go to the Logger screen to view the options.

2. Press **Ctrl+Esc**. The application server appears in the menu as **exteNd Application Server**.

📖 For more information, see "Using startup options" below and "Specifying the JVM to use" on page 72.

Using startup options

There are two kinds of startup options you can provide on the command line:

Type of startup options	How you specify them
Options passed directly to the JVM (Java Virtual Machine) or handled by the SilverServer executable to launch the JVM	Using the plus (+) sign
Application server-specific options passed to the class that starts the server	Using the minus (-) sign

What the server does

When passing the options that were specified with the plus (+) sign to the JVM, the application server changes the plusses to minuses for processing by the JVM. For example, if you specify the following command line:

```
SilverServer +verbose -dbcheck
```

The equivalent command line is:

```
java -verbose ServerStartupClass -dbcheck
```

Displaying +options

To see a list of possible options for the JVM, type the following commands:

Command	Description
java -?	Lists standard options.
java -X	Lists nonstandard options. Note that these options are subject to change without notice.

NOTE: The application server automatically adds the following option with the appropriate value to the command line: **-Djava.class.path**. This option will override any corresponding option you specify on the command line.

Table of options

These are the server startup options:

Server startup option	Description
Supported Java options: `+<x>` (These options are passed to the JVM. For more information about the Java + options, see your Java documentation.)	
+client	(Windows only) Application server–specific option. Use the client HotSpot JVM. 📖 For more information, see "Specifying the JVM to use" on page 72.
+cp:a *path*	Appends specified *path* to the class path. This option makes additional Java classes available to applications by appending the specified path to the class path. NOTE: Use the AGCLASSPATH environment variable to extend Java classes. For more information, see "Setting the AGCLASSPATH variable" on page 96.
+cp:p *path*	Prepends specified *path* to the class path. Don't use this debugging option without first contacting Novell exteNd Technical Support. Instead, use AGCLASSPATH to make additional Java classes available to applications. For more information, see "Setting the AGCLASSPATH variable" on page 96.
+debug	Application server–specific option. You must set this option to debug server-side objects.
+Djava.compiler=none	Application server–specific option. You must set this option to profile server-side applications.
+profile	Application server–specific option. You must set this option to profile server-side applications.
+server	(Windows only) Application server–specific option. Use the server HotSpot JVM. 📖 For more information, see "Specifying the JVM to use" on page 72.
+verbose[:class | gc | jni | vmopts]	Run the JVM verbosely. There is an application server-specific option for +verbose: `+verbose:vmopts` Specifying this option tells the server to output the startup options to the console without all the other output generated in verbose mode.
+Xms *size*	Initial Java heap size within the JVM. Default value is 16 MB. NOTE: See the override information in the next row.
+Xmx *size*	Maximum Java heap size within the JVM. Default value is 256 MB. You can override +Xms and +Xmx. For example, if you are running a development server that services only one user, you might want to run the server with a lighter memory footprint using the following command line: `SilverServer +Xms2m +Xmx16m` This sets the initial Java heap size to 2 MB and the maximum heap size to 16 MB.

Server startup option	Description
Application server options: `-<x>` (These options are passed to the application server.)	
-? or -help	Print usage for SilverServer.exe.
--a	Print the server startup properties, then exit without starting the server. This debugging option is useful if the server fails to start. You can see what the startup properties are.
-host *hostname*	The full name of the host running the server. Not required unless there are problems with hostname resolution.
-jvmversion	Print information about the JVM.
-minspan *number*	Use with -retry *number* (see below). The duration in minutes within which the retries must be made. SilverMonitor ceases to operate after the number of minutes specified by minspan, even if not all retry attempts have been made. The default value is 10. 📖 For more information, see "Using SilverMonitor" on page 223.
-dbcheck	Checks database integrity at server startup.
-noexitondbcheck	Don't exit if the database integrity check fails. Use this option to check integrity and allow access to the SMC if the database check fails.
-nomonitor	Run without the SilverMonitor background program. This option is useful for debugging the server when it fails to start. If the option is not used, the server will keep trying to start. 📖 For more information, see "Using SilverMonitor" on page 223. NOTE: If you start the server with -nomonitor, you will not be able to restart the server from the SMC (or using the API). You will have to shut down the server and then restart it again manually.
-p *file*	Read startup properties from a specified file. Defaults to the server's **\Resources\httpd.props** file.
-retry *number*	The number of times SilverMonitor should attempt to restart the server or process before ceasing to operate. The default value is 3. See -minspan *number* (above). 📖 For more information, see "Using SilverMonitor" on page 223.
-trace	Turn tracing on. Dump tracing information to the default or specified log output.

Specifying the JVM to use

The HotSpot JVM shipped with the application server comes (on most platforms) in two versions: a client version and a server version. This section describes how the server-side processes (the server, Cache Manager, Load Manager, and Dispatcher) and clients (such as SilverJ2EEClient and the SMC) use these JVMs.

On Windows

On Windows, by default the server-side processes and clients all use the server version of the HotSpot JVM.

To override this behavior on Windows, you can use the following startup options with any of the application server's executables that start a JVM:

Executable	JVM it uses
`+server`	Server HotSpot JVM
`+client`	Client HotSpot JVM

On UNIX and Linux

On UNIX and Linux, the JVM usage is different. These are the defaults:

Platform(s)	JVM usage
Solaris and Linux	Server processes use the server HotSpot JVM. Clients use the client HotSpot JVM.

To override this behavior for UNIX and Linux server-side processes, edit the server's .agprofile file. Look for the case statement and update the definition of the native search path for your platform (LD_LIBRARY_PATH) to point to the JVM you want.

On NetWare

On NetWare, by default the server-side processes and clients all use the client HotSpot JVM.

➢ **To change the JVM:**

1. Stop all running Java applications.
2. Shut down the JVM by typing:

   ```
   java -exit
   ```
3. Choose **one** of these methods for changing to the server JVM:
   - Type `load java -server`
   - Set the JAVA_COMPILER environment variable:

     ```
     env JAVA_COMPILER=server
     ```
   - Set the JAVA_COMPILER in the sys:\etc\java.cfg file:

     ```
     JAVA_COMPILER=server
     ```
4. Restart Java by typing:

   ```
   load java.nlm
   ```
5. Restart the application server.
6. Restart other Java applications as desired.

Starting the server on a specific IP address or hostname

You can set the **http-server.com.sssw.srv.host** property in the httpd.props file (located in the server's **Resources** directory) to direct the application server to start on a specific IP address or hostname. This feature is particularly helpful on machines with multiple network cards and multiple IP addresses (*multihoming*). This feature works identically on Windows and UNIX.

For example:

```
http-server.com.sssw.srv.host=192.101.1.10
```

Shutting down the application server

Use the server **Stop** button on the SMC (see procedure below) to shut down the resident or selected server when you are taking the machine out of service or if you need to install a software patch.

NOTE: If you want to stop and restart the server in order to activate properties you have modified, use the **Restart** button, as described in "Restarting the application server" on page 74.

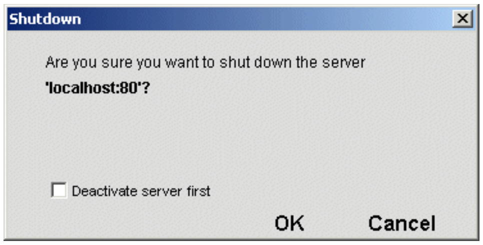
➢ **To shut down a server:**

1 Start the SMC.

2 Select the server you want to stop from the left panel.

3 Click **Stop**.

The following confirmation displays:



Shutdown

Are you sure you want to shut down the server
'localhost:80'?

□ Deactivate server first

4 (Optional) Select **Deactivate server first** if you want the server deactivated before it is shut down or restarted (see table below for more information).

5 Select **OK**.

What happens next depends on whether or not you selected **Deactivate server first**:

Situation	Result
Deactivate server first is not selected	The server is immediately shut down or restarted.
Deactivate server first is selected	No new client sessions can be established, but existing client sessions continue to operate normally. In clusters, the deactivated server is unregistered from the Load Manager so new sessions will not be dispatched to the server (if you are using a third-party load manager, there is no way of notifying it that the server is deactivated). Once the last client session is closed (typically five minutes after the last client connection is closed), the server is declared deactivated and is shut down or restarted. NOTE: The SMC is a client connection to the server, so you must either exit the SMC or remove the server from the SMC before the server can shut down.

Restarting the application server

Using the **Restart** button is the recommended way to stop and restart a server to update server property changes you have made using the SMC.

You can restart a server only if it was started with SilverMonitor (the default). For more information, see "Using SilverMonitor" on page 223.

➢ **To restart an application server:**

1. Start the SMC.
2. Select the application server from the left panel.
3. Click **Restart**. You are prompted to confirm the restart.
4. (Optional) Select **Deactivate server first** if you want the server deactivated before it is restarted (see table below for more information).
5. Select **OK**.

   What happens next depends on whether or not you selected **Deactivate server first**:

Situation	Result
Deactivate server first is not selected	The server is immediately restarted.
Deactivate server first is selected	No new client sessions can be established, but existing client sessions continue to operate normally. In clusters, the restarted server is unregistered from the Load Manager so that new sessions will not be dispatched to the server (if you are using a third-party load manager, there is no way of notifying it that the server is deactivated). Once the last client session is closed (typically five minutes after the last client connection is closed), the server is declared deactivated and is restarted. NOTE: The SMC is a client connection to the server, so you must either exit the SMC or remove the server from the SMC before the server can restart.

The application server restarts using the same startup parameters as when it was originally started and picks up any changes you have made to the server's properties.

Maintaining processes running as services

Instead of manually starting the application server, you can run it as a service (or as a daemon in UNIX), so that it starts automatically when the server machine reboots. For information on installing the server to run as a service, see *Installing Novell exteNd*.

In addition to running the server as a service, you can also run the following server-side processes as services:

- Load Manager
- Dispatcher
- Cache Manager

To help you manage processes running as Windows services, the server provides **SilverServiceUtil**, a Windows utility that allows you to:

- Create, list, delete, and stop application server and non–application server services
- Define additional Windows services for application server processes
- Create new console log files each time an application server service is restarted
- Define command-line arguments that will be automatically passed to an application server process when it starts as a Windows service
- Reconfigure an existing application server process when it is running as a service

NOTE: To initially get the server set up to run as a service, you should use the installation program, as described in *Installing Novell exteNd*. After you have the server running as a service, you can use SilverServiceUtil to maintain your service environment, including creating additional services on the same machine.

Using SilverServiceUtil

SilverServiceUtil is a command-line utility in the server's **\bin** directory.

➢ To invoke SilverServiceUtil:

- Change to the **\bin** directory and enter:

  ```
  SilverServiceUtil
  ```

  You will see usage notes.

The utility has the following actions:

Action	Description
addDepend	Add a dependency to the service (if a service is dependent on another service, the Windows service manager won't start it until the other service has started)
create	Create a new service
delete	Delete an existing service
list	List all services defined on the current machine
stop	Stop a running service
update	Update the configuration of an existing application server service

These actions are described next. For complete information about using each of the SilverServiceUtil actions, enter the following:

```
SilverServiceUtil -action -help
```

Defining a dependent service

➢ To make a service dependent on another service:

- Enter the following:

```
SilverServiceUtil -addDepend -service serviceName -prereq prereqServiceName
```

where:

Option	Description
serviceName	The name of the service that is dependent on *prereqServiceName*
prereqServiceName	The name of the service that *serviceName* is dependent on

When this dependency has been defined, the Windows service manager will not start *serviceName* until *prereqServiceName* has been started.

Creating a service

You can use SilverServiceUtil to create an application server service or a non–application server service.

Creating an application server service You can use SilverServiceUtil to configure all installed application servers to run as a service.

➢ **To create an application server service:**

- Enter the following:

```
SilverServiceUtil -create -service serviceName -display displayName -program pathToExecutable
[-outputDir outputDirectory -maxOutputFiles numFiles -startupOptions options]
```

where:

Option	Description
serviceName	The name of the service to create. The name is arbitrary but must be unique.
displayName	The display name of the service. The name is arbitrary but must be unique.
pathToExecutable	Path to the executable to invoke for the service. Specify one of the following executables (all are in the server’s **\bin** directory): ◆ **SilverAppServerService.exe** to run the server as a service ◆ **SilverCacheManagerService.exe** to run the Cache Manager as a service ◆ **SilverDispatcherService.exe** to run the Dispatcher as a service ◆ **SilverLoadManagerService.exe** to run the Load Manager as a service If you are configuring multiple servers on a single host to run as services, make sure you point to unique executables each time you execute SilverServiceUtil.
outputDirectory	(Optional) Path to the directory in which to save log files. If not specified, the log files are saved in the server’s temp directory.

numFiles	(Optional) Maximum number of log files to create in *outputDirectory*.
	If you specify 0, or if the value is not specified, the log file will be overwritten each time the service is restarted. The log file will be named:
	`nameOfService.out`
	For example, if you are running a Version 4 server as a service named SilverAppServerService4 and have specified 0 for *numFiles*, the one and only log file would be named:
	`SilverAppServerService4.out`
	If you specify a number larger than 0, a new file will be created each time the service is restarted, up to the number you specify, after which the numbering restarts. The log files will be named:
	`nameOfService.nnn.out`
	Using the preceding example, the first log file would be named:
	`SilverAppServerService4.000.out`
	The next time the service starts, the following log file will be created:
	`SilverAppServerService4.001.out`
	A warning message is sent to the server console when 80 percent of *numFiles* has been reached. If *numFiles* itself is reached, the service will behave as if *numFiles* were defined as 0, until the output files are removed.

options	(Optional) Command-line options to pass to the executable when it is started. Enclose the options in double quotation marks. The specific options depend on the executable.

Creating a generic service You can also use this option to create a generic Windows service.

➢ To create a generic service:

- Enter the following:

```
SilverServiceUtil -create -service serviceName -display displayName -program pathToExecutable -generic
```

Listing and deleting services

You can list all services defined on the current machine as well as delete any existing service.

➢ To list all services:

- Enter the following:

```
SilverServiceUtil -list [-d]
```

If you specify -d, the display name will be listed along with the service name.

➢ To delete a service:

- Enter the following:

```
SilverServiceUtil -delete -service serviceName
```

Specify the service name, not the display name. (You are **not** asked to confirm the deletion.)

Stopping a service

➢ To stop a service:

- Enter the following:

```
SilverServiceUtil -stop -service serviceName [-retries numRetries -delay retryDelay]
```

where:

Option	Description
serviceName	The name of the service to stop
numRetries	(Optional) Number of times to query the service manager to determine whether the server has stopped; if not specified, the service manager is not queried
retryDelay	(Optional) Seconds between retry attempts

Reconfiguring a service

➢ To reconfigure an existing application server service:

- Enter the following:

```
SilverServiceUtil -update -service serviceName
[-outputDir outputDirectory -maxOutputFiles numFiles -startupOptions options]
```

where the -outputDir, -maxOutputFiles, and -startupOptions arguments are the same as used in the create action.

The update action modifies the Windows registry entry for the corresponding service. The changes won’t take effect until the service is restarted. You can start and stop services through the Windows Services control panel (without rebooting the machine).

Setting up separate ports

To restrict access to specific types of application server operations, you can define the following ports:

Port	Description
Runtime	Allows users to run J2EE applications using HTTP, HTTPS, or RMI.
Administration	Allows administrators to set or modify administration settings such as the ability to read and write server settings, security, certificates, and so on. The administration port is required to: ◆ Run the SMC ◆ Use SilverCmd ◆ Make server administration API calls

Each port type excludes URLs and operations that are not associated with it. For example, the administration port only passes administration URLs. The separate ports are designed to work in conjunction with your server permission settings. For example, if your administration and runtime ports are unique, any attempt to run an administration URL on the runtime port will fail. Once a user successfully accesses an administration port, the server checks the user's group permissions to further determine the level of access.

How you configure a public site would probably be different from the way you would configure an e-commerce site that uses credit card transactions. Particularly in an extranet environment, you don't want users attempting to perform administration operations that alter your application data. Configuring multiple server ports in conjunction with your corporate firewall lets you manage internal and external access to your applications.

Using separate ports with your firewall

There are several security advantages to defining separate application server ports for different types of users and operations:

- Opening a single runtime port through your firewall for users outside your organization limits your security risk.
- Separating administration and runtime port access helps prevent unauthorized users from attempting to administer the server. Users connecting to your applications may know the runtime port number, but only administrators need be aware of the administration port.
- Defining an administration port that can only be accessed from inside the firewall helps restrict calls made to that port.

About enabling ports

The server supports administrative and runtime ports for each of these protocols:

Protocol	Default port
HTTP	83 on NetWare
	8080 on UNIX
	80 on Windows
HTTPS (RSA)	443
HTTPS (DSA)	443

By default, only the HTTP port is enabled. The DSA and RSA ports are set to the default values, but not enabled. The server is not listening on the DSA and RSA ports until you enable them.

When the application server starts, it binds a socket to each unique port value you have configured and enabled. The application server does not require unique port values for the different types of access; ports having the same value will share the same socket and allow multiple operations. For example, if you set your HTTP runtime and administration ports to 8080, the application server will use only one socket to accept requests for both.

TIP: When you install the application server, the HTTP runtime and administrative ports are configured to whatever port number you specify as the default. You will need to update your program shortcut used to launch the SMC if you configure a separate administration port after you install the application server.

For alternative ways to start the SMC, see “Running the SMC” on page 25.

Clients connecting to an administration port can perform only operations associated with that port. Because many objects involved with administration require runtime support, runtime operations can be performed on any port. However, runtime ports only allow runtime operations.

For how to enable HTTP ports, see “Specifying general server properties” on page 80. For more information about enabling HTTPS ports, see “Enabling RSA/DSA ports” on page 149.

Port types

The application server can be set to a maximum of six unique port numbers for HTTP/HTTPS communications. The type of operations a port allows and its associated security protocols can be configured independently—that is, you can mix any of the three security protocols with any of the three port types:

Connection access type	Connection port type	Port property name	Default port
Unencrypted port using HTTP	Runtime	com.sssw.srv.port_rt	83 in NetWare, 8080 in UNIX, 80 in Windows
	Administrative	com.sssw.srv.port_admin	
SSL port using RSA encryption	Runtime	com.sssw.srv.https.port_rsa_rt	443
	Administrative	com.sssw.srv.https.port_rsa_admin	
SSL port using DSA encryption	Runtime	com.sssw.srv.https.port_dsa_rt	
	Administrative	com.sssw.srv.https.port_dsa_admin	

All port property names (as defined in the https.props file) begin with **http-server**. For more information, see Appendix A, “The httpd.props File”.

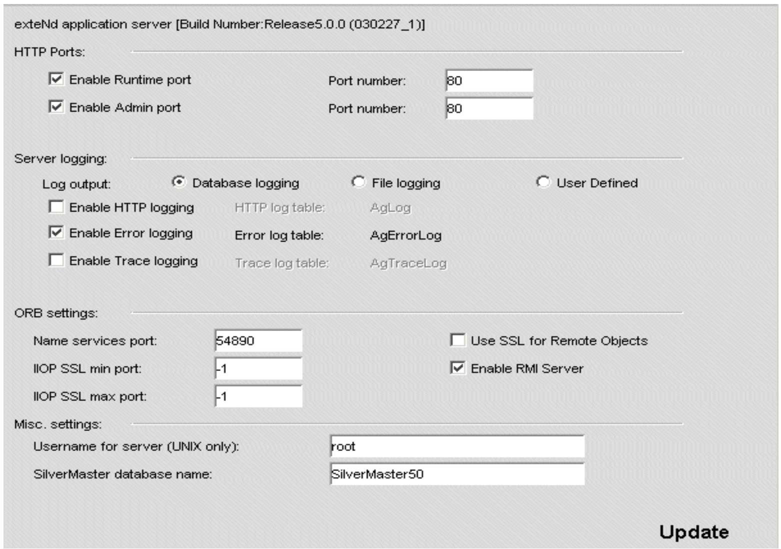
Specifying general server properties

General server properties include:

- HTTP listener ports
- Name of the SilverMaster database
- User account that starts the server on UNIX

➢ To specify general server properties:

1. Start the SMC.
2. Select the server.
3. Select the **Configuration** icon from the toolbar.
4. Select **General**:



exteNd application server [Build Number:Release5.0.0 (030227_1)]

HTTP Ports:

☑ Enable Runtime port — Port number: 80

☑ Enable Admin port — Port number: 80

Server logging:

Log output: ● Database logging ○ File logging ○ User Defined

□ Enable HTTP logging — HTTP log table: AgLog

☑ Enable Error logging — Error log table: AgErrorLog

□ Enable Trace logging — Trace log table: AgTraceLog

ORB settings:

Name services port: 54890 — □ Use SSL for Remote Objects

IIOP SSL min port: -1 — ☑ Enable RMI Server

IIOP SSL max port: -1

Misc. settings:

Username for server (UNIX only): root

SilverMaster database name: SilverMaster50

5. Edit any of these fields as needed:

Field(s)	What to specify
Enable HTTP **runtime** port and Port number Enable HTTP **Admin** port and Port number	To enable HTTP listener ports, select any or all of the HTTP port options and then specify a corresponding port number. By default for all HTTP port types, the application server listens on port: ◆ 83 in NetWare ◆ 8080 in UNIX ◆ 80 in Windows The default port for HTTPS (RSA) and HTTPS (DSA) is 443. 📖 For more information, see “Setting up separate ports” on page 79. 📖 To disable HTTP communications, see “Turning off HTTP communications” on page 150.
Username for server (UNIX only)	To specify the user under whose account the server is started on UNIX. The default is **root**.
SilverMaster database name	To change the SilverMaster database the server uses. For example, you might need to change the name of the SilverMaster database if you are setting up a load balancing cluster. All servers in a cluster must use the same SilverMaster database. You’d use this field to specify the **SilverMaster** database name.

6. Click **Update**.
7. To activate the changes, click **Restart**.

📖 For more information, see “Restarting the application server” on page 74.

Using server logging

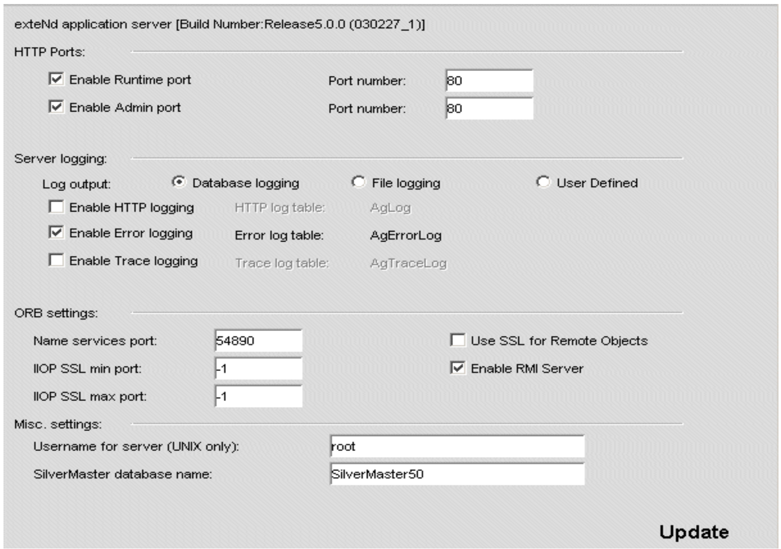
The application server provides logging for things like server debugging, server monitoring, and security auditing. You can log the information to a file or a database, or you can specify your own custom class to perform the logging.

➤ **To turn on logging:**

1 Start the SMC.

2 Select the **Configuration** icon from the toolbar.

3 Select **General**:



exteNd application server [Build Number:Release5.0.0 (030227_1)]

HTTP Ports:

☑ Enable Runtime port — Port number: 80

☑ Enable Admin port — Port number: 80

Server logging:

Log output: ● Database logging ○ File logging ○ User Defined

☐ Enable HTTP logging — HTTP log table: AgLog

☑ Enable Error logging — Error log table: AgErrorLog

☐ Enable Trace logging — Trace log table: AgTraceLog

ORB settings:

Name services port: 54890 — ☐ Use SSL for Remote Objects

IIOP SSL min port: -1 — ☑ Enable RMI Server

IIOP SSL max port: -1

Misc. settings:

Username for server (UNIX only): root

SilverMaster database name: SilverMaster50

4 Select the logging option(s) you want to turn on or off as follows:

Field	Description	Usage
Database logging	Logs messages to SilverMaster. Messages are stored in the AgLog, AgErrorLog, and AgTraceLog system tables.	This is the default setting.
File logging	Logs messages to files that you specify.	Specify the file name for each option you activate in the text field next to the option (which is enabled if either **File logging** or **User Defined** is selected).
User Defined	Uses a custom Java class to do the logging.	By default, the application server uses its own internal class to do the logging. If you want to customize the log output—for example, to specify an extended log file format—you can write your own logging class and then specify it here. To learn how to create and use a custom logging class, see Chapter 13, “Using the Server Administration API”.

Field	Description	Usage
Enable HTTP logging	Writes a line in the AgLog table (or file you specify) for every client request to the server and every server response.	Run in conjunction with error logging. Use standard HTTP logging when you want to see client requests to the server and also when you want to monitor server activity. Use in conjunction with the **Statistics/Summary/Request time** option in the SMC (see "Summary statistics" on page 106). 📖 For more information, see "About HTTP logging" on page 83.
Enable Error logging	Records errors and miscellaneous status information in the AgErrorLog table (or the file you specify). If you enable this type of logging, you get more detailed information about server errors and status.	Turn this option on.
Enable Trace logging	Records server actions. Unlike HTTP logging and error logging, trace logging concentrates on tracking server events as well as error messages. When enabled, the AgTraceLog table (or the file you specify) contains additional tracing information that Technical Support uses to track down server issues.	Turn this option on only if Technical Support requests it.

5 Click **Update**.

The application server starts logging the specified information.

About HTTP logging

When you enable HTTP logging, by default the server logs HTTP messages in the standard W3C common log file format (see www.w3.org) to the database. You can redirect the log to a file as described above.

There is also a compound log file format, which is like the common log file format except that it also logs the Referrer (which allows click tracing) and User-Agent (which logs the browser type) fields from each HTTP request. The application server provides built-in support for the compound log file format.

➢ To log using this format:

1 In the SMC, under Server logging select **User Defined**.

2 In the **Java class** field, specify the following:

```
com.sssw.srv.http.CompoundLogger
```

3 Specify the file to log the HTTP requests to (and optionally, files for error and trace logging, also supported by the CompoundLogger class).

4 Restart the server.

TIP: Instead of using the SMC, you can also specify compound logging by setting these values in httpd.props and restarting the server:

```
http-server.com.sssw.srv.logger=com.sssw.srv.http.CompoundLogger
http-server.com.sssw.srv.logger.logname=fileName
```

Viewing the log

If you are using the built-in logging class, you can view the log in the SMC (see “Displaying logs” on page 103). You can also use the SilverCmd PrintLog to view the log in the SilverCmd console window or a file. PrintLog allows you to view the data regardless of whether you are using the built-in logging class or the database.

Maintaining log tables and files

Log information can expand quickly. Clean out your log tables and log files to keep them manageable. You can use SilverCmd ClearLog to delete records. Use your native database utilities to maintain these tables, or use any editor to shorten or delete extraneous information from log files.

Specifying ORB settings

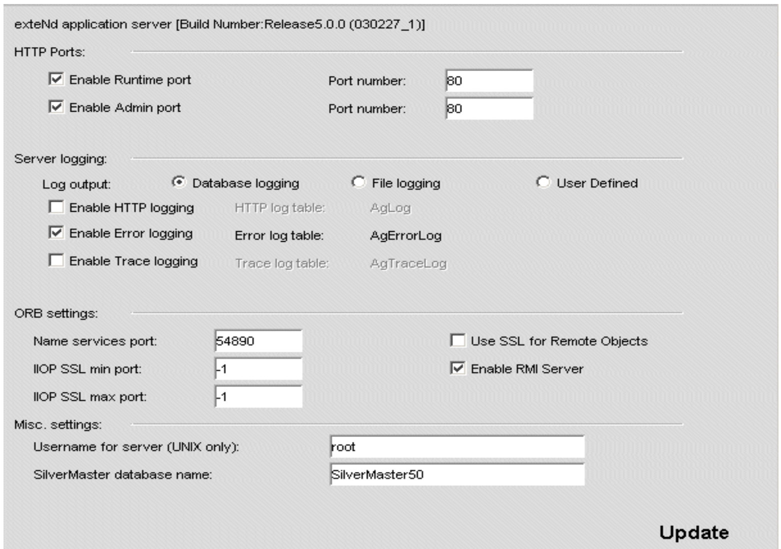
You can use the SMC to specify whether to use RMI—and if so its name services port, whether to use SSL for the remote objects, and the ports to use for IIOP SSL.

➢ **To specify ORB settings:**

1 Start the SMC.

2 Select the **Configuration** icon from the toolbar.

3 Select **General**:



exteNd application server [Build Number:Release5.0.0 (030227_1)]

HTTP Ports:

☑ Enable Runtime port   Port number: 80

☑ Enable Admin port   Port number: 80

Server logging:

Log output: ● Database logging ○ File logging ○ User Defined

□ Enable HTTP logging   HTTP log table: AgLog

☑ Enable Error logging   Error log table: AgErrorLog

□ Enable Trace logging   Trace log table: AgTraceLog

ORB settings:

Name services port: 54890   □ Use SSL for Remote Objects

IIOP SSL min port: -1   ☑ Enable RMI Server

IIOP SSL max port: -1

Misc. settings:

Username for server (UNIX only): root

SilverMaster database name: SilverMaster50

4 Specify the RMI options:

Field	Description
Name services port	The port the application server starts the RMI name services on (for example, all clients that need to find an EJB use this service). The default is 54890.
Enable RMI Server	Specifies whether to enable use of RMI for unencrypted client communications. You can enable RMI separately or together with HTTP. If selected, the application server exports a remote server object using RMI/IIOP and accepts RMI sessions, so that non-HTTP clients don't require an HTTP session on the server. If this option is not selected, the RMI server is not created and does not accept RMI sessions. NOTE: To encrypt a remote transaction, enable the **Use SSL for Remote Objects** option.
Use SSL for Remote Objects	Specifies whether to use SSL encryption to secure the RMI Server (if enabled), remote sessions, and the remote user transaction. If selected, remote objects (such as EJBs) can be encrypted and exported by non-HTTP clients using RMI/IIOP.
IIOP SSL min port	Specifies the lower bounds (in a range) for IIOP SSL communications. If you do not specify a range, the ORB picks the first available port. Use –1 when you do not need to specify a range. You **must** create an IIOP SSL port range: ◆ To allow interoperability with network firewalls. Controlling the range used by IIOP SSL communications allows the firewall administrator to open these ports and configure traffic appropriately. ◆ When your environment supports session-level failover for EJBs using IIOP SSL communications. The range must be large enough to allow one port for each unique combination of EJB security attributes used by the beans deployed on your system. The maximum number of combinations of security attributes (and therefore the largest range) is 64 (when using the standard set of cipher suites). A range of 16 is a reasonable number of ports for most common installations.
IIOP max port	Specifies the upper bound (in a range) for IIOP SSL communications. If you specify –1, there is no upper bound to the range.

5 Click **Update**.

6 To activate the changes, click **Restart**. For more information, see "Restarting the application server" on page 74.

Running multiple servers on one host

You can run more than one application server on a host at one IP address. (The application server also supports *multihoming*, where you have multiple IP addresses through multiple network cards on one host. See "Starting the server on a specific IP address or hostname" on page 73).

Specifying unique ports Multiple servers running on one host must be configured to use unique ports. You can specify runtime and administration ports in the SMC:

Ports	Default	Information on setting
HTTP	80	"About enabling ports" on page 79
RSA	443	"Enabling RSA/DSA ports" on page 149
DSA		
RMI name services	54890	"Specifying ORB settings" on page 84
SSL IIOP port	–1	

Specifying character set encoding

The application server uses the following server property when URL-encoding and -decoding form content:

```
com.sssw.srv.international.UrlEncoding
```

The encoding property is stored in the **AgUserIni.props** configuration file, which is located in the server's **Resources** directory.

By default, the application server uses utf-8 (Universal Character Set Transfer Format) for URL encoding and decoding. Because utf-8 can encode ASCII characters without requiring modification, utf-8 works well for English and most other Western languages. Because languages using multibyte encodings are not a subset of utf-8, character encoding and decoding will not work properly with them.

When to change the encoding scheme

You typically need to change the encoding scheme only when the majority of client browsers in your environment use character encodings that are not ISO 8859-1 (Latin 1). For example, a Japanese Web site that serves content to its employees using the ShiftJIS encoding may want to change the encoding property to SJIS.

➢ **To change from the default utf-8 to another encoding:**

1 Add the following line to the AgUserIni.props file (located in the **Resources** directory below the application server's root directory):

```
com.sssw.srv.international.UrlEncoding=NewEncoding
```

2 Enter the language mapping needed at your site in place of the *NewEncoding* variable. If you are unsure about the Java string mapping for your language, check the Sun Web site.

3 Restart the application server.

URL content will be encoded using the new encoding scheme **after** you restart the server.

Running the JMS server

The application server includes the Novell exteNd JMS server for its JMS (Java Message Service) implementation. That means the exteNd JMS server provides the JMS server that runs with the application server to support messaging in your J2EE applications.

This section describes some things you need to know about using the Novell exteNd JMS server with the application server:

- Starting the JMS server
- Using JMS servers in clusters
- Displaying JMS server debug messages

To learn more about the JMS server, see the Novell exteNd Messaging Platform help.

Starting the JMS server

You can start the JMS server in either of two ways:

Method	How it works
Automatic	When the application server starts, it can check for the JMS server and automatically start it if necessary. To use this approach, you need to make sure the application server's httpd.props file specifies this property setting: `http-server.com.sssw.srv.jmsServerLaunch=true` When you install the application server, the installation program asks if you want to configure JMS server and then sets this property according to your response. If you later change your mind, you can edit the httpd.props file yourself to specify `true` or `false`. By default, the installation program sets the jmsServerLaunch property to `true`. But note that this property defaults to `false` if you remove it from the httpd.props file. In the automatic approach, the application server launches the JMS server as a **child process**. As a result, the JMS server will terminate when the application server terminates.
Manual	You can start the JMS server yourself, as described in the Novell exteNd Messaging Platform help. If you start the JMS server manually before starting the application server, the application server won't try to start the JMS server (regardless of the jmsServerLaunch property setting).

Using JMS servers in clusters

You can increase the reliability of JMS servers in your environment through the use of clusters. Here are some common configurations:

Configuration	What you do
Clustered application servers with individual JMS servers	Set up a cluster of application servers following the instructions in Chapter 12, “Administering a Cluster”. By default, each application server in the cluster has its own local JMS server.
Clustered application servers accessing clustered JMS servers	**1** Set up a cluster of application servers following the instructions in Chapter 12, “Administering a Cluster”. **2** Set up a cluster of JMS servers following the instructions in the Novell exteNd Messaging Platform help. You’ll need to manually edit the **msgsvc.properties** file installed for each application server (look in the JMS server **\lib** directory).

Displaying JMS server debug messages

You can troubleshoot JMS server–related problems at runtime by displaying debug messages to the application server console. To turn on basic JMS server debugging, edit the application server’s httpd.props file to specify this property setting:

```
http-server.com.sssw.srv.jms.debug=1
```

For more detailed JMS server debugging, specify a number greater than 1 for this property. To disable JMS server debugging, specify 0 (the default).

6 Setting Up Users and Groups

This chapter describes how to define **Silver Security** users and groups—users and groups known only to the Novell exteNd Application Server. It contains these sections:

- About Silver Security users and groups
- Managing Silver Security users and groups
- Using the Locksmith privilege

NOTE: The application server also provides access to external security providers, including Windows, LDAP, NIS+, and certificate issuers. For information about setting up access to users and groups from these providers, see "Accessing security provider systems" on page 124.

About Silver Security users and groups

You can define Silver Security users and groups in many ways. For example, you might want to define groups based on your site's organization—such as Accounting, Sales, and so on—and assign users to those groups. The groups can contain Silver Security users as well as users defined in external security realms. Users can belong to multiple groups.

After you define Silver Security users and groups, you can define access to any directories or objects in the system based on the Silver Server users and groups. For example, you might want to set certain permissions for members of the Accounting group and other permissions for members of the Developers group.

📖 For more information about using users and groups to set data permissions, see "Authorization and access control" on page 165.

Two predefined groups After installation, the application server provides two predefined groups: Administrators and Developers. Both groups initially contain only the server administrator. Use these groups as a starting point for creating your own users and groups. If you want to use names that differ from the predefined group names, you can rename and then delete them. For more information, see "Managing Silver Security users and groups" on page 90.

Group	Description
Administrators	After installation, the server administrator is the only member of this group. This person is initially the only one with the Locksmith privilege (which includes the ability to add new users and groups). See "Using the Locksmith privilege" on page 94. Add any users that have to perform administration tasks to this group. You can assign to users in this group all or a subset of administration permissions. To administer the server, users need to be assigned Modify Server Configuration access. See "Administrative server permissions" on page 166.
Developers	After installation, the only privilege users in this group have (compared to users not part of the Administrators group) is the ability to browse directory listings.

Case sensitivity Silver Security user names and passwords are case sensitive as follows:

- **User names** follow the SilverMaster database: they are case-insensitive if the SilverMaster database is case-insensitive (for example, **administrator** and **Administrator** are considered the same name), and they are case-sensitive if SilverMaster is case-sensitive.
- **Passwords** are always case-sensitive. For example, **admin** and **Admin** are always considered different passwords.

📖 For more information, see "Default group permissions" on page 172.

About your administrator account

Your administrator account can be assigned to any user recognized by the application server (Silver Security, Windows, LDAP, NIS+, or Certificate user).

When you installed the application server, you specified the user name and password for the application server administrator account. This account was used when the new SilverMaster database catalog was created.

You use the server administrator account to log in to the SMC to administer the application server. You also need to specify the server administrator account to run some of the SilverMasterInit command-line options.

The server administrator user account is part of the predefined Administrators group and has the Locksmith privilege. The Locksmith privilege provides Set Permissions privileges to any object on the server. Only accounts with the Locksmith privilege are able to assign Locksmith privilege to another account.

📖 For more information, see "Using the Locksmith privilege" on page 94.

NOTE: The **server** administrator account, which restricts who can log in and administer the application server, is distinct from the **database** administrator account. The application server uses the database administrator account when connecting to the SilverMaster database. The only time you need to specify the SilverMaster database account is when you are running SilverMasterInit at the command line.

➢ **To create a new administrator account:**

1. Log in to the SMC using the existing Administrator account.
2. Create a new administrator account or select an existing user from one of the security realms to be the administrator.
3. Click **Properties** and assign the new account **Locksmith** privilege.
4. Add the new administrator account to the **Administrators** group.
5. Close the SMC.
6. Restart the SMC and log in as the new administrator.
7. Verify (using the **Properties** dialog) that the new account has **Locksmith** privilege.
8. (Optional) Delete the older Administrator account.

Managing Silver Security users and groups

You can use the SMC to add Silver Security users, edit user properties, and add Silver Security groups.

NOTE: You can also perform these tasks using SilverCmd. For more information, see SetUserGroupInfo in the SilverCmd reference chapter of the *Facilities Guide*.

Adding Silver Security users

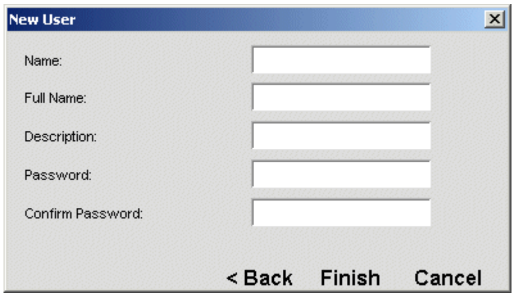
To add a user:

1. Start the SMC.
2. Select the **Security** icon from the toolbar.
3. Select **Users & Groups**.
4. Expand **Silver Security** and select **Users**.
5. Choose the **Add New User** icon at the bottom of the right pane:

   You are asked whether you want to define a Silver user or a certificate user.
6. Select **Silver user** and click **Next**.

   For information on defining certificate users, see "Manually installing client certificates" on page 155.

   The New User panel displays:



7. Type the appropriate information in each field.

   The Name field specifies the short name for the user. This is the name the user types in the Login box.
8. After completing the panel, click **Finish**.

Editing user properties

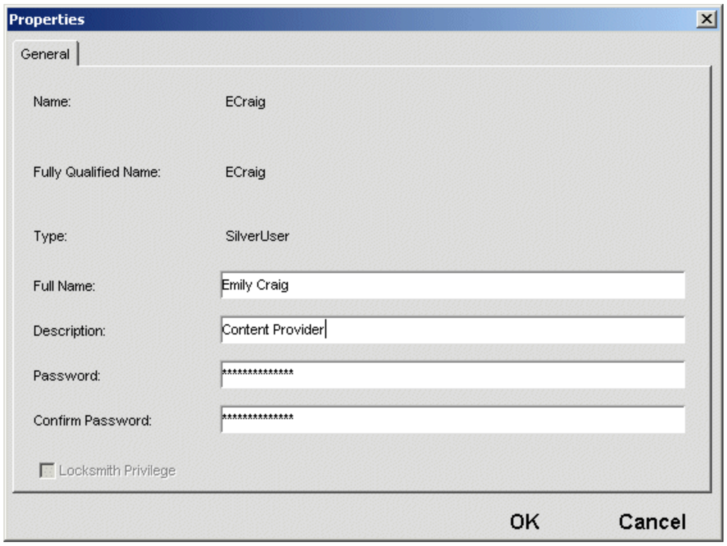
You can use the SMC to change user properties. (For users defined in external security providers, the only editable property is the Locksmith privilege; for more information, see "Using the Locksmith privilege" on page 94.)

Not allowing users to modify their properties By default, users can change their own user properties. You can turn off this privilege. For more information about this privilege, see "Enabling authentication" on page 159.

To edit user properties:

1. Start the SMC.
2. Select the **Security** icon from the toolbar.
3. Select **Users & Groups**.
4. Expand the **Silver Security** list of users.
5. Highlight a user name and choose **Properties**.

The following panel displays:



Name: ECraig

Fully Qualified Name: ECraig

Type: SilverUser

Full Name: Emily Craig

Description: Content Provider

Password: **************

Confirm Password: **************

□ Locksmith Privilege

6 Modify any of the four editable fields.

The Fully Qualified Name field corresponds to the Name field used to create the user and is not editable.

If you have Locksmith privilege, you can also change whether the user you are modifying has Locksmith privilege.

For more information, see “Using the Locksmith privilege” on page 94.

7 Click **OK**.

Adding Silver Security groups

Creating groups helps streamline security administration by allowing you to categorize users within a larger context, such as a business organizational unit or a work role. A user can belong to one or more user groups, and can be granted access to objects by group or individual status.

➤ To create a group:

1 Start the SMC.

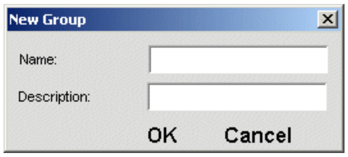
2 Select the **Security** icon from the toolbar.

3 Select **Users & Groups**.

4 Expand **Silver Security** and select **Groups**.

5 Choose the **Add new Silver Security group** icon:

The following panel displays:



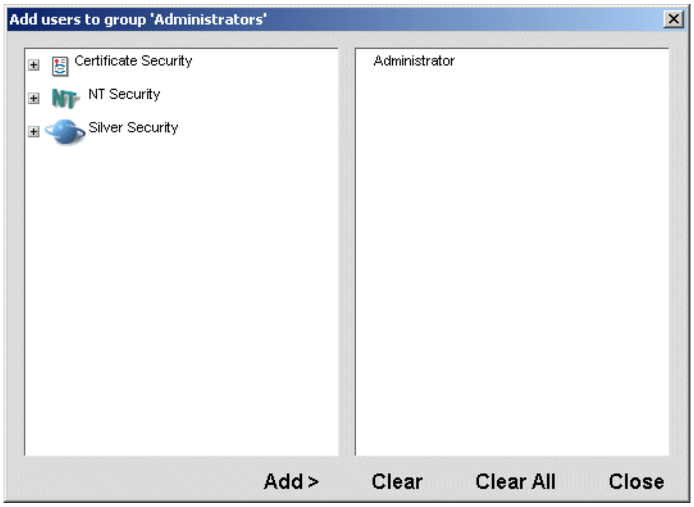
6 Enter a name and a description for the group.

7 Click **OK**.

➢ To add users to a group:

1. Start the SMC.
2. Select the **Security** icon from the toolbar.
3. Select **Users & Groups**.
4. Expand **Silver Security** and expand **Groups**.
5. Select the Silver Security group to which you want to add users.
6. Choose the **Add user to group** icon:

   The following panel displays:

   NOTE: Your panel may look different depending on which external security providers you have configured and the operating system used by the application server. For more information, see “Accessing security provider systems” on page 124.

7. To add a user to the group, select the user in the left panel, then choose **Add**.

   You can add users defined by external security providers to Silver Security groups.

   - To remove a selected user, choose **Clear**.
   - To remove all users in the group, choose **Clear All**.

8. When finished, click **Close**.

Using the Locksmith privilege

The Administrator user has the Locksmith privilege by default. The Locksmith privilege allows users to do the following:

Task	More information
Get and set data access permissions even if these permissions are denied elsewhere in the system (for example, if the user does not belong to a group for which set permission is allowed).	📖 For information about defining security for the server and objects on the server, see "Authorization and access control" on page 165.
Read server property settings from the SilverMaster database, even if this permission is denied elsewhere in the system	Since the Locksmith privilege also allows setting permissions, Locksmiths can also give themselves server administrative permissions. NOTE: Locksmiths don't have all permissions just by virtue of being Locksmiths. But as Locksmiths they can give themselves any permissions they want.
Grant and revoke the Locksmith privilege for other users	See "Editing user properties" on page 91.

NOTE: Since the Locksmith privilege provides powerful access to server functions and properties, limit the Locksmith privilege to trusted users.

Keep at least one Locksmith Be careful not to delete all users with the Locksmith privilege: a user must have Locksmith privilege to grant it to someone else. So if no one has Locksmith privilege, it cannot be granted.

If you find yourself in that situation, you can run SilverMasterInit with the –l command-line option to define a Locksmith account.

📖 For more information, see "Using the SilverMasterInit program" on page 224.

7 Maintaining the Server

This chapter describes how to perform some typical maintenance tasks on the Novell exteNd Application Server. It has sections on:

- Administering an application server remotely
- Setting the AGCLASSPATH variable
- Maintaining deployed J2EE objects
- Managing J2EE transactions
- Monitoring server activity
- Integrating with existing Web servers

Administering an application server remotely

You can use the SMC to remotely administer servers. You can administer as many servers as you want from one SMC console. If you are running a server in a cluster, you can choose additional server clusters to administer.

📖 For more information about server clusters, see “Server clustering” on page 195.

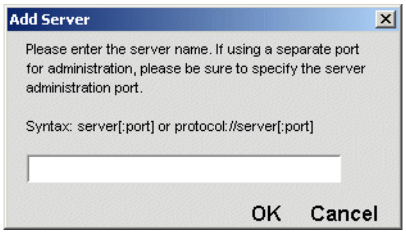
➢ **To administer a server:**

1 Make sure the server you want to administer is up and running.

2 Start the SMC.

3 Select the **Choose** (server) icon.

The Add Server dialog displays:



Add Server

Please enter the server name. If using a separate port for administration, please be sure to specify the server administration port.

Syntax: server[:port] or protocol://server[:port]

4 Specify ***server:port***, where:

Parameter	Description
server	The name of the server (such as localhost or http://*hostname*).
port	The administration port. You need to specify the port only if it is not the default port for your operating system. 📖 For more information about ports used by the server, see “Setting up separate ports” on page 79.

5 Click **OK**.

6 Use the SMC to administer the server.

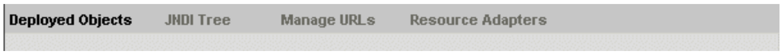
Setting the AGCLASSPATH variable

The application server supports an environment variable called **AGCLASSPATH**, which allows you to extend the Java classes available to your applications. You can use AGCLASSPATH, for example, when you want to include third-party elements such as database drivers with your application. Use this variable instead of the **+cp** Java class path option described in "Starting the application server" on page 69. The application server overrides the CLASSPATH variable.

Operating system	Description
NetWare	The application server starts via an NLM, so you must set AGCLASSPATH using the **setenv** environment with semicolon-separated components; for example: `setenv AGCLASSPATH=path1;path2;path3;...;pathx`
UNIX	Set the environment variable AGCLASSPATH following the appropriate procedure for the shell you are using.
Windows	Set the AGCLASSPATH environment variable using the System settings tool available via the Control Panel.

Maintaining deployed J2EE objects

The Deployment Options section of the SMC provides access to information about the J2EE objects that have been deployed on the server:



The following table describes the functionality for each of the panels:

Use this panel	To perform this function
Deployed Objects	Manage J2EE deployed applications like EARs, WARs, EJB JARs, application client JARs, and RARs.
JNDI tree	View the RMI JNDI (Java Naming and Directory Interface) tree for any application server currently managed by the SMC or to view the InitialContext of any server available to the SMC via the network.
Manage URLs	Use this panel to specify a default page for a database or server. **Database URL**—The page that displays when the user requests the URL for the database, such as http://localhost/MyApp/. If there is no default database page defined, the user sees a directory listing for the database (if this is allowed by the administrator). **Server URL**—The page that displays when the user requests the URL for the server, such as http://localhost/. If there is no default server page defined, the user sees a directory listing for the server (if this is allowed by the administrator).
Resource Adapters	View information about deployed resource adapters and their settings.

Using the Deployed Objects panel

The **Deployed Objects** panel lists:

- **EJB JAR** files containing EJBs
- **WAR** files containing JSP pages and servlets
- **EAR** files that package other archive files into a full application
- J2EE application-client deployments

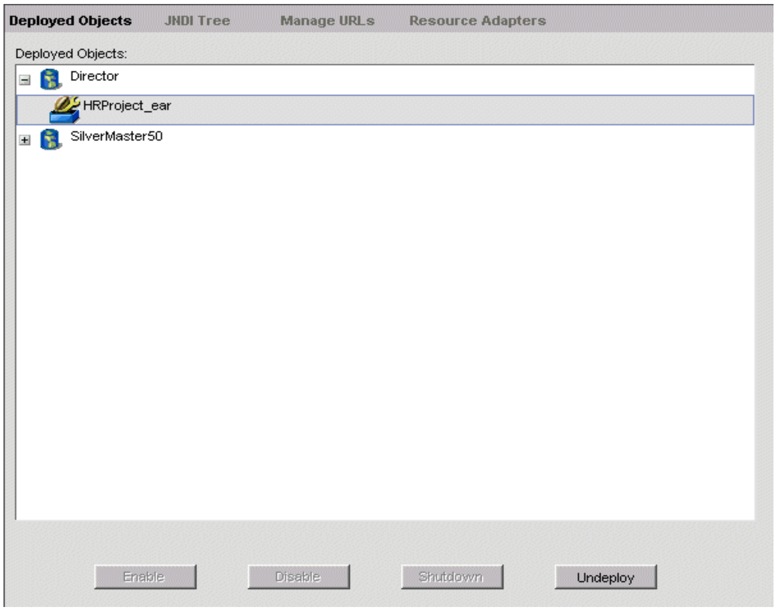
📖 For more information, see the J2EE deployment chapter in the *Facilities Guide*.

➤ To manage deployed J2EE objects:

1. Start the SMC.
2. Select the **Deployment** icon from the toolbar.
3. Select **Deployed Objects**.
4. Expand the database containing the deployed objects you want to manage:

   You can **Undeploy** WARs, EARs, CARs, and RARs.

   You can **Enable**, **Disable**, **Shutdown**, and **Undeploy** EJB JARs.



5. Select a deployed object and perform one of the following actions.

Action	Description
Enable	For EJB JARs only. Enables a disabled EJB JAR. If a JAR is enabled, the beans in the JAR are available.
Disable	For EJB JARs only. Disables an enabled EJB JAR. If a JAR is disabled, none of the beans in the JAR is available. Disabling a JAR stops any running EJBs in the JAR. A disabled JAR remains disabled until it is explicitly enabled.

Action	Description
Shut Down	For EJB JARs only. Shuts down the JAR and all its beans for the current server session. When a JAR is shut down, none of the beans in the JAR is available. When the server is restarted, enabled JARs that had been shut down in the previous server session are again available.
Undeploy	Shuts down and removes deployed objects from the server.

➢ **To view deployed RARs:**

1. Start the SMC.
2. Select the **Deployment** icon from the toolbar.
3. Select **Resource Adapters**.

   The deployed resource adapters are displayed in the dropdown.
4. Select a resource adapter from the list to display the adapter’s settings.

➢ **To view the JNDI tree:**

1. Start the SMC.
2. Select the **Deployment** icon from the toolbar.
3. Select **JNDI tree**.
4. Choose the radio button for the function you want to perform:

Radio button	Description
RMI	Displays the RMI JNDI tree on the selected server.
Specify URL	Displays the InitialContext for any server available to the SMC via the network. To view the InitialContext for an LDAP server named beetle, you would enter the following: `ldap://beetle/dc=novell.com`

5. If you chose Specify URL, click **Submit**.

➢ **To specify the default database URL:**

1. Start the SMC.
2. Select the **Deployment** icon from the toolbar.
3. Select **Manage URLs**.
4. Select the **Database URL** radio button.
5. Choose the database whose default page you want to set from the Database dropdown.
6. Enter an URL in the URL text box and click **set Default URL**.

   The URL should be a database-relative URL.

➢ **To specify the default server URL:**

1. Start the SMC.
2. Select the **Deployment** icon from the toolbar.
3. Select **Manage URLs**.
4. Select the **Server URL** radio button (it is selected by default).
5. Enter an URL in the URL text box and click **set Default URL**.

The URL should be a server-relative URL and should include the database name (the same database you chose from the Database dropdown in the procedure above).

If your applications are deployed to the SilverMaster database, you do not need to include a database name.

Managing J2EE transactions

The application server supports J2EE transactions via the Novell exteNd Transaction Manager (TM), which is part of the Novell exteNd Messaging Platform. The TM is the transaction service for the ORB. It provides an implementation of the JTA TransactionManager and UserTransaction interfaces. These interfaces represent the contract between the transaction manager and the application server, and between the transaction manager and user applications.

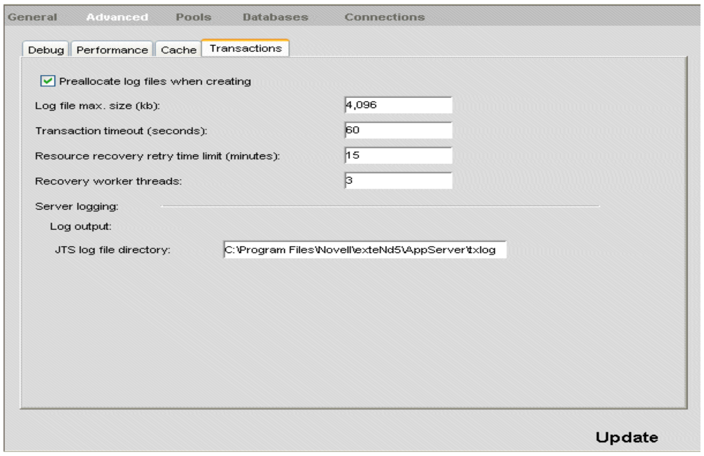
You can use the SMC to specify settings for the TM transaction log, the resources the TM uses to recover transactions, and so on.

How the TM recovers transactions Transactions need to be recovered when the server suffers a catastrophic error and needs to be restarted. Here is the process the TM performs at server restart:

1. The TM reads the transaction log file.
2. The TM determines that a transaction needs to be recovered when items in the log have been prepared for a transaction but the transaction did not complete (in other words, the transaction did not commit or roll back).
3. The TM creates a number of worker threads (you specify the number via the SMC, as described in the procedure below). These worker threads are used to recover the incomplete transactions.
4. If a transaction includes access to a remote resource (like a CORBA resource), the worker threads will attempt to access those same resources. If the worker threads are unable to locate a remote resource, the worker thread will sleep for a specified length of time (the **Resource recovery retry time limit** specified in the SMC) before it attempts to access the resource again. In the meantime, the other recovery worker threads work on other incomplete transactions in the log.

➢ **To specify Transaction Manager settings:**

1. Start the SMC.
2. Select the **Configuration** icon from the toolbar.
3. Select **Advanced**.
4. Select the **Transactions** tab:



☑ Preallocate log files when creating

Log file max. size (kb): 4,096

Transaction timeout (seconds): 60

Resource recovery retry time limit (minutes): 15

Recovery worker threads: 3

Server logging:

Log output:

JTS log file directory: C:\Program Files\Novell\exteNd5\AppServer\txlog

5 Specify the settings as follows:

Field	Description
Preallocate log files when creating	Specifies whether to preallocate the transaction log file.
Log file max size (kb)	The maximum size of the transaction log file in KB. When the maximum file size is reached, the TM attempts to create a new log file.
Transaction timeout (seconds)	The amount of time allowed for all transactions managed by the TM to complete. The timer begins when the transaction starts. If the transaction does not complete before the transaction timeout period ends, the TM rolls back the transaction. Developers can override this value for specific transactions using the UserTransaction interface.
Resource recovery retry time limit (minutes)	Specifies the amount of time a worker thread will sleep before it attempts to access the remote resource again after a worker thread is unable to obtain a remote resource during a transaction recovery.
Recovery worker threads	The number of threads the TM should create to process a log file during a recovery. The higher the number of recovery worker threads the faster the recovery. However, since there is unlikely to be a large number of transactions to recover, the recovery will probably not take a lot of time. These threads are destroyed after the recovery completes.
JTS log file directory	The disk location for the transaction log file.

6 Click **Update**.

7 To activate the new settings, click the **Restart** button.

Monitoring server activity

The SMC provides several options that allow you to monitor server activity easily. These options are described in the following sections:

- Displaying charts of server activity
- Displaying logs
- Displaying views of server statistics

Displaying charts of server activity

You can display real-time charts of various server statistics.

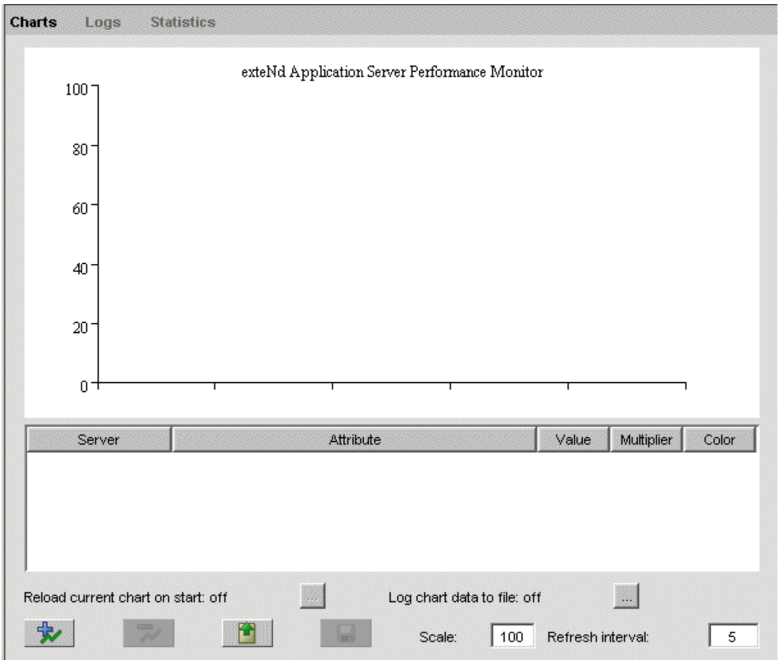
➢ **To display a chart of server activity:**

1 Select the server or cluster you are administering.

2 Select the **Monitor** icon from the toolbar.

3 Select **Charts**.

An empty chart displays:



Server	Attribute	Value	Multiplier	Color

Reload current chart on start: off

Log chart data to file: off

Scale: 100

Refresh interval: 5

4 Click **Add Plot**.

The Add Plot dialog displays, allowing you to choose which statistics you want to chart. The statistics are divided into categories.

5 Select a statistic you want to chart and click **Add**. Statistics marked with ¹²³ are count values; statistics marked with △ record changes in values.

The statistic is added to the table below the chart. The current value of the statistic is shown, as is the color of the plot line that will be generated.

6 (Optional) Select additional statistics, clicking **Add** after each one.

7 When you have selected all the statistics you are interested in, click **Close**.

What happens The application server plots all statistics you have specified and displays the exact values in the table below the chart. By default, the values are refreshed every five seconds. (To change this, enter a new number in the Refresh Interval box.)

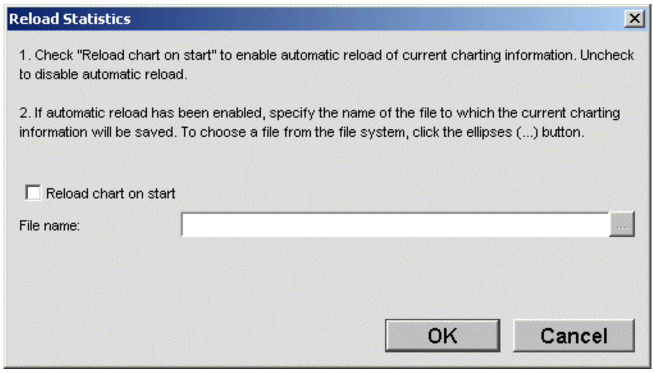
Saving statistics settings for automatic reload

By default, the statistics you are plotting are not saved and reloaded between sessions of the SMC.

➢ **To save statistics settings:**

1 Click the **Reload current chart on start** button at the bottom of the Charts panel.

The Reload Statistics dialog displays:



Reload Statistics

1. Check "Reload chart on start" to enable automatic reload of current charting information. Uncheck to disable automatic reload.

2. If automatic reload has been enabled, specify the name of the file to which the current charting information will be saved. To choose a file from the file system, click the ellipses (...) button.

☐ Reload chart on start

File name:

2 Check the **Reload chart on start** check box.

3 Specify a file name or choose the ellipsis to choose a file from the file system.

4 Click **OK**.

The charting data is saved to an XML file that is read each time you restart the SMC.

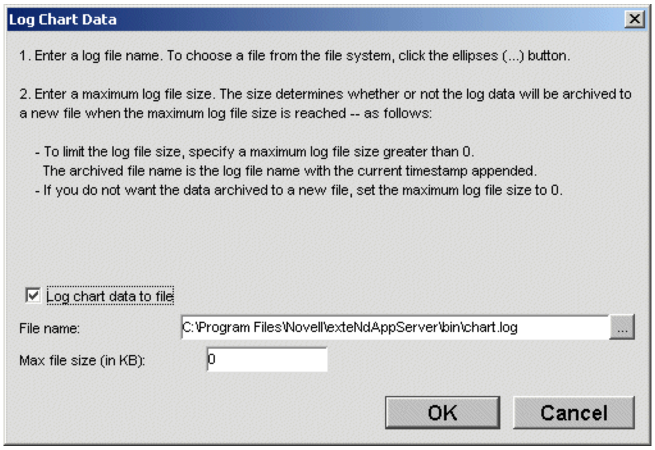
Saving chart data

By default, the statistics you chart are not saved.

➢ **To save chart data:**

1 Click the **Log chart data to file** button at the bottom of the Charts panel.

The Log Chart Data dialog displays:



Log Chart Data

1. Enter a log file name. To choose a file from the file system, click the ellipses (...) button.

2. Enter a maximum log file size. The size determines whether or not the log data will be archived to a new file when the maximum log file size is reached -- as follows:

- To limit the log file size, specify a maximum log file size greater than 0. The archived file name is the log file name with the current timestamp appended.
- If you do not want the data archived to a new file, set the maximum log file size to 0.

☑ Log chart data to file

File name: C:\Program Files\Novell\exteNdAppServer\bin\chart.log

Max file size (in KB): 0

2 Check the **Log chart data to file** check box.

3 Specify a file name or choose the ellipsis to choose a file from the file system.

4 Specify the log file size or specify 0.

If you specify a log file size: when that file size is reached, the data is dumped to a file of the same name—but with the timestamp appended (to make the files unique).

If you specify 0: there is no limit on how large this file may grow (except the limits imposed by the file system).

5 Click **OK**.

The data is saved to a tab-delimited file.

Changing the scale and refresh rate

You can change the scale of the Y-axis by specifying a new value in the **Scale** field (the default is 100). You can also change how often the application server updates the values by changing the value in the **Refresh interval** field (the default is every five seconds).

Removing a plot

➢ **To remove a plot:**

- Select a plot and click **Remove plot**.

Editing a plot

It may be that you are plotting different statistics with very different scales but want them to appear clearly in the chart. To do this, you can change the multiplier for particular plots to equalize the values and make the chart easier to read.

➢ **To edit a plot:**

1. Select **the multiplier column** in the row of the statistics you want to edit.
2. Select **a multiplier value** from the dropdown.

Saving a statistics set

Once you have charted a set of statistics that you are interested in, you can save the specification of the statistics set in a file so that you can easily view the set of statistics later.

NOTE: The file stores the list of statistics that are plotted but does not store the statistics' values.

➢ **To save a statistics set:**

1. Display the statistics you want to save as a set.
2. Click **Save**.

   The Save dialog displays.
3. Specify the file you want to save the statistics set in. The default extension is XML.
4. Click **Save**.

   You can view the statistics set later by clicking **Load**, as described next.

Viewing a statistics set

➢ **To view a statistics set you have saved:**

1. Click **Load**.

   The Open dialog displays.
2. Select the file that defines the statistics set you want to view and click **Open**.

Displaying logs

If you have enabled server logging and are using the built-in logging class to log to the database, you can display the log(s) in real time in the SMC. (If you are logging to a file or using a custom class to do the logging, you can't display the log in the SMC.) As an alternative, you can use the PrintLog SilverCmd.

📖 For more information about server logging, see "Using server logging" on page 82.

➢ **To display a log:**

1. Select the server or cluster you are administering.
2. Select **Monitor** options.
3. Select the **Logs** panel.
4. Select the tab corresponding to the logging you have enabled and want to view.

What you can do You can:

- Update the log by clicking **Refresh** (the log does not update automatically)
- Sort the log by clicking a column header
- Resize a column by placing the mouse pointer on the column's right border and dragging the mouse
- View the message by double-clicking on the row or clicking **Log Detail**

Displaying views of server statistics

You can display in the SMC tabular views of server statistics related to individual sessions and threads, as well as a summary of server activity.

➢ **To access server statistics in a view:**

1. Select the server or cluster you are administering.
2. Select the **Monitor** icon from the toolbar.
3. Select **Statistics**.
4. Select the tab for the category you want.

   Statistics are updated dynamically.

What you can do You can:

- Force an update by clicking **Refresh**
- Resize a column by placing the mouse pointer on the column's right border and dragging the mouse

About the statistics The sections that follow describe the statistics that are displayed.

Session statistics

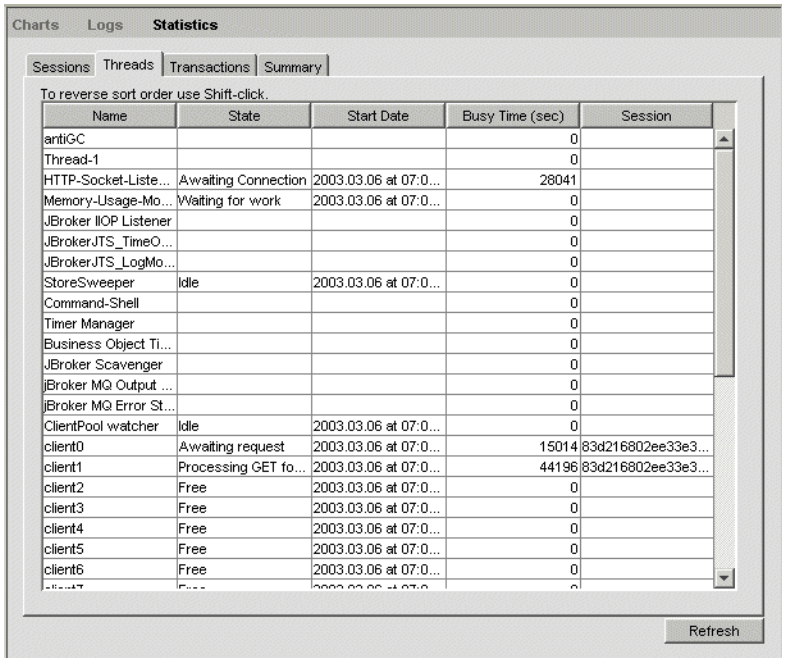
This tab displays statistics for each current client session:

Session statistic	Description
ID	Displays the session ID returned by a call to the AgiSession internal system table.
User name	Displays the user name of the person or entity logged on to this session. If unknown, displays **Anonymous**.
State	Displays the state of the connection.
Logged in	Displays **true** (checked) if the user has logged in (for example, from the browser's login). Otherwise, displays **false** (unchecked).
Host	Displays the host of the client source, if known.
Idle time	Displays the time elapsed (in seconds) since the last client connection on this session ended.
Protocol version	Displays the application server protocol used for the session.

Viewing statistics in a browser You can also display these statistics in a browser. Point your browser to http://*server*/SilverStream/Sessions.

Thread statistics

This tab displays statistics for each server thread:



To reverse sort order use Shift-click.

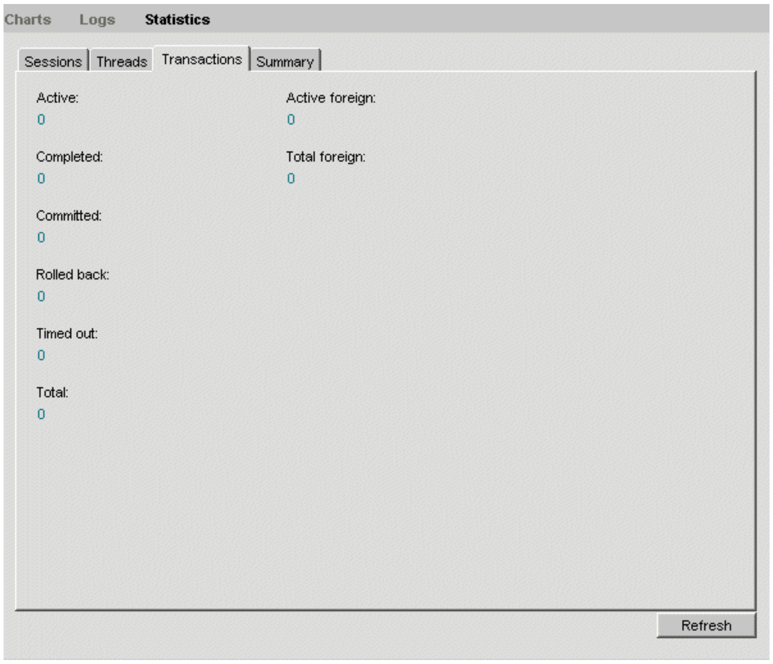
Name	State	Start Date	Busy Time (sec)	Session
antiGC			0	
Thread-1			0	
HTTP-Socket-Liste...	Awaiting Connection	2003.03.06 at 07:0...	28041	
Memory-Usage-Mo...	Waiting for work	2003.03.06 at 07:0...	0	
JBroker IIOP Listener			0	
JBrokerJTS_TimeO...			0	
JBrokerJTS_LogMo...			0	
StoreSweeper	Idle	2003.03.06 at 07:0...	0	
Command-Shell			0	
Timer Manager			0	
Business Object Ti...			0	
JBroker Scavenger			0	
jBroker MQ Output ...			0	
jBroker MQ Error St...			0	
ClientPool watcher	Idle	2003.03.06 at 07:0...	0	
client0	Awaiting request	2003.03.06 at 07:0...	15014	83d216802ee33e3...
client1	Processing GET fo...	2003.03.06 at 07:0...	44196	83d216802ee33e3...
client2	Free	2003.03.06 at 07:0...	0	
client3	Free	2003.03.06 at 07:0...	0	
client4	Free	2003.03.06 at 07:0...	0	
client5	Free	2003.03.06 at 07:0...	0	
client6	Free	2003.03.06 at 07:0...	0	

Following is a description of each of the fields:

Thread statistic	Description
Name	**Nonclient** threads are used for various internal tasks (such as cleaning up server data structures). **Client** threads handle incoming requests. There should be at least as many client threads as there are Maximum number of client connections listed in the SMC Connections panel (see "Client connection parameters" on page 183).
State	A brief description of the current state of the thread.
Start date	The start date for the thread. In many cases this date is the same as when the server was started. But for dynamically allocated threads this value will be different.
Busy time	The elapsed time (in seconds) since the thread has been actively working, as opposed to waiting. This value reflects how busy the server is in general.
Session	An internal session ID for this thread. See the Sessions tab to determine user/host information.

Transactions statistics

This tab provides statistics for transactions managed by the Novell exteNd Transaction Manager (TM):



Active:
0

Completed:
0

Committed:
0

Rolled back:
0

Timed out:
0

Total:
0

Active foreign:
0

Total foreign:
0

Following is a description of each of the fields:

Transactions statistic	Description
Active	The number of active transactions this server is responsible for managing.
Completed	The number of completed transactions.
Committed	The number of committed transactions.
Rolled back	The number of transactions rolled back.
Timed out	The number of transactions timed out.
Total	The sum of the active, rolled back, and completed transactions (excludes foreign transactions).
Active foreign	The number of active foreign transactions. A *foreign transaction* is a transaction that was started in another process (such as a different server or client application) that was propagated to this server via call to an EJB. The foreign transaction is controlled by the other process.
Total foreign	The sum of the active, rolled back, and completed foreign transactions.

Summary statistics

This tab provides access to different types of category summaries. The following describes the items for each tab selection:

- **Server statistics**

Server statistic	Description
Server load	Provides a scaled value based on overall server activity.
Total number of hits since server was last started	The number of HTTP message requests the server has received since it was started.
Date/time server was started	The date and time the server was started.
Number of bytes emitted by server	The number of bytes returned by the server since it was started.

- **Request time statistics** The statistics for this tab are enabled only when HTTP logging is enabled. For a description of HTTP logging, see "Using server logging" on page 82.

Request time statistic	Description
Minimum request processing time	The minimum elapsed time to process any HTTP request, from when the server receives the header until it transfers the reply.
Minimum request URL	The URL of the request that took the least amount of processing time.
Maximum request processing time	The maximum elapsed time to process any HTTP request, from when the server receives the header until it transfers the reply.
Maximum request URL	The URL of the request that took the most processing time (see previous row).
Mean request processing time	The mean (average) of the processing times for all requests received by the server.

- **Memory statistics** The statistics for this option apply to the Java Virtual Machine (JVM).

Memory statistic	Description
Free memory	The result for the Java call Runtime.freeMemory().
Total memory	The result of the Java call RunTime.totalMemory().
GC count	The total number of times the Java Garbage Collector has run since the server was started. This number indicates how hard the server is being hit, how many objects are allocated, and how much memory pressure the server is under.

- **Thread statistics** This option provides statistics about client threads that handle incoming client connections.

Thread statistic	Description
Free thread count	The current number of threads not associated with a client connection and available for immediate use.
Idle thread count	The number of threads associated with a client connection but not currently handling a user request.
Total thread count	The total number of client threads allocated. This number should be equal to the **Maximum number of client connections** in the SMC Connections panel (see "Client connection parameters" on page 183).

Viewing statistics in a browser You can also display most of these summary statistics in a browser. Point your browser at http://*server*:*port*/SilverStream/Statistics. The page updates itself automatically every five seconds.

Integrating with existing Web servers

The application server provides **Web server integration (WSI) modules** that allow you to redirect requests for selected pages served by your Web server to an application server. You can use these WSI modules to integrate a Novell exteNd Application Server with your Web server.

📖 For more information, see Chapter 8, “Using the Web Server Integration Modules”.

8 Using the Web Server Integration Modules

The Novell exteNd Application Server provides **Web server integration (WSI)** modules that allow you to integrate the application server within your existing Web server framework. This chapter includes these topics:

- About the WSI modules
- Customizing your WSI configuration
- Security considerations for IIS and iPlanet

About the WSI modules

The WSI modules are provided as an enhancement to the Novell exteNd Application Server. The WSI modules are not included when you install the Novell exteNd Application Server from the product CD. If you want to use the WSI modules, you must obtain them from the Novell developer Web site at: developer.novell.com/ndk/wsi.htm. Install and configure them as instructed.

For NetWare users The Apache WSI is automatically installed and configured during NetWare installation. The NetWare installation generates a configuration file called AgWSI.conf that includes a basic Apache WSI setup; Apache's httpd.conf file references the AgWSI.conf file. So you do not need to download the WSI modules from the developer Web site.

How the WSI modules work

A WSI module extends the URL namespace of your existing Web server directory structure by forwarding requests for a specific URL (or set of URLs) to an application server that services the requests. It works like this:

1. When a WSI module receives a request for a specified URL, it opens an HTTP connection to an application server and forwards the request to it.
2. The application server services the request and returns the response to the WSI.
3. The WSI then returns the response to the Web server.

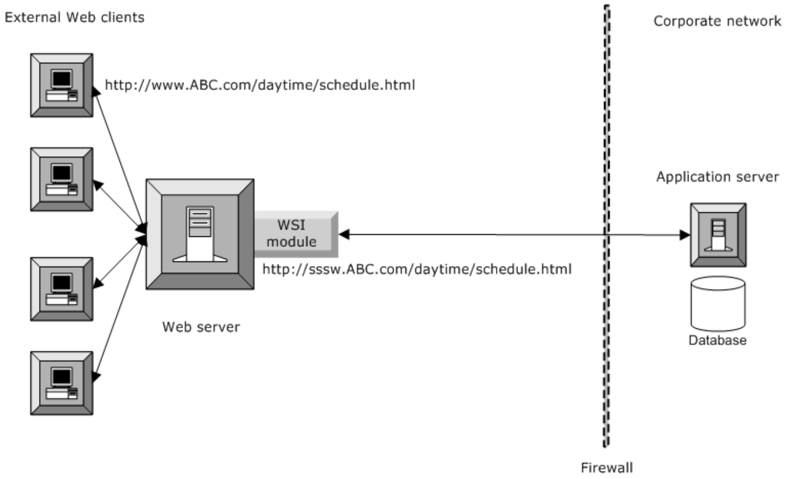
There is no direct communication between the browser and the application server; all calls pass through the WSI. To improve response time, you can configure the WSI to pool connections to the application server.

WSI module sample configurations

The following five examples show how a WSI might be used in a corporate network.

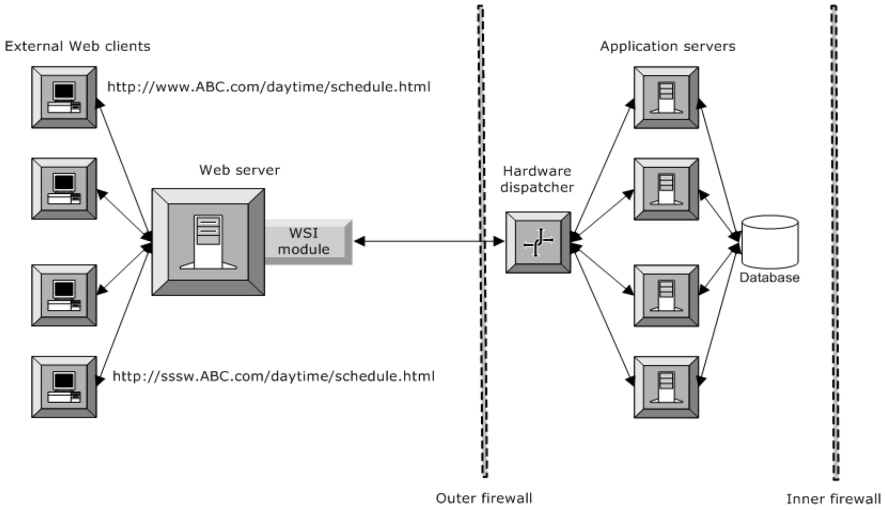
- In the first four examples, the WSI module forwards a client request for the URL http://**www.ABC.com**/daytime/schedule.html to http://**sssw.ABC.com**/daytime/schedule.html (the URL address of the client does not change).
- The final example shows a configuration where different client requests are forwarded to different application servers.

Example 1: Web server with one application server In this simple scenario, the WSI module forwards requests from a Web server to a single application server:

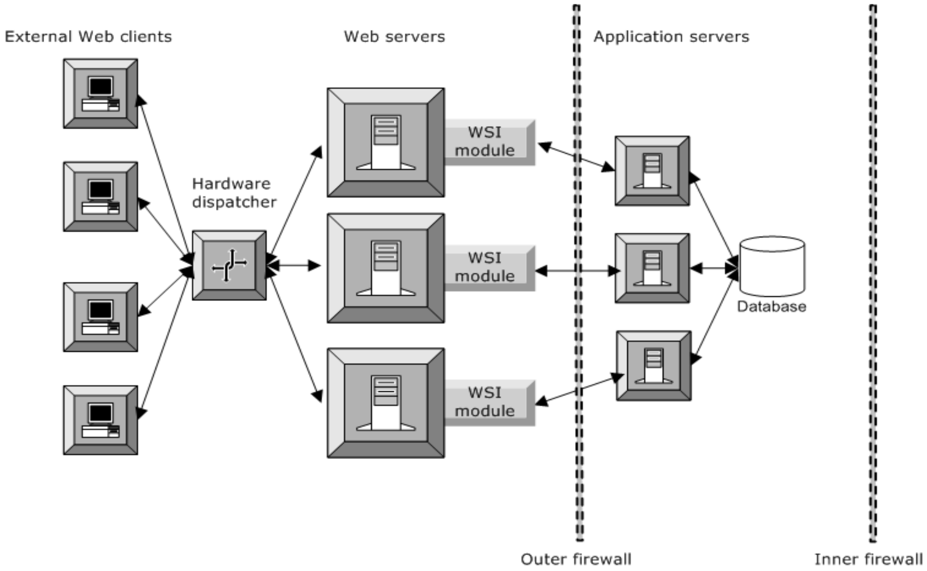


NOTE: There are many ways to set up your environment. In each of the following four examples, for example, you could position your database behind the inner firewall. Positioning the corporate database outside the DMZ helps protect your database.

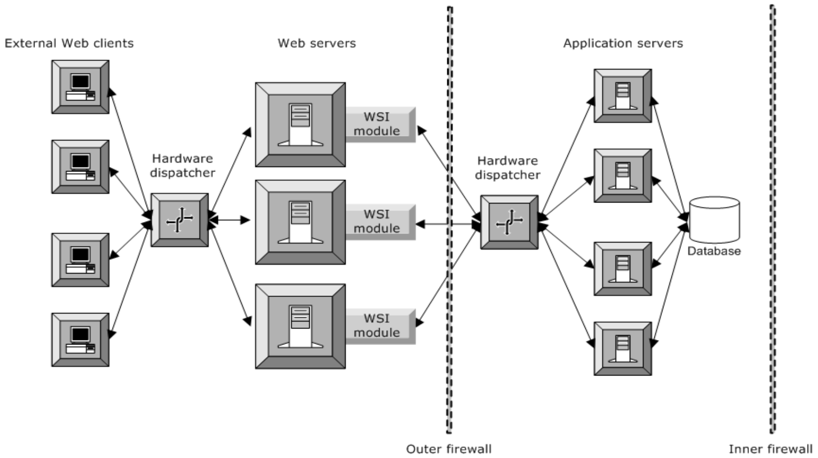
Example 2: Web server with clustered application servers Here the WSI module forwards requests from a single Web server to a cluster of application servers (this approach increases database access and reduces the risk of the application server becoming a single point of failure):



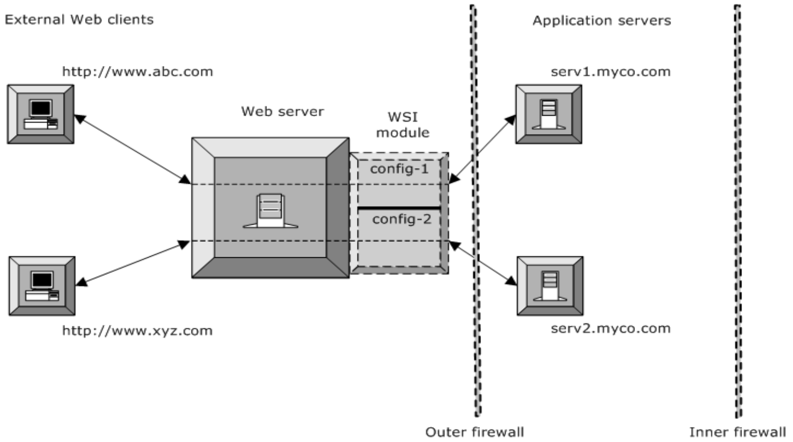
Example 3: Multiple Web servers and application servers Here multiple WSI modules are configured to forward requests to multiple application servers. This approach is more reliable for handling a large volume of requests:



Example 4: Hardware dispatcher for load balancing Here again multiple WSIs forward requests from multiple Web servers to multiple application servers. Now the largest volume of requests can be more reliably processed—because of multiple servers at the front and back ends and the use of dispatcher load balancing at both ends:



Example 5: Web server with multiple application servers Here a WSI module is set up with multiple configurations, allowing requests to be routed to different application servers depending on the incoming URL. Requests for abc.com are directed to serv1.myco.com, and requests for xyz.com are directed to serv2.myco.com. For more information, see "Directing requests to multiple application servers" on page 113.



WSI configuration requirements

Before you install, configure, and enable the WSI modules, note that:

- The WSI will **not** work with application servers configured to require client certificates for authentication.

  The WSI module cannot forward a client certificate from the browser to the application server. Because the WSI opens a new HTTPS connection to the application server, the WSI would need to access the private key of the client certificate to use the client certificate for this connection. However, the private key is not available—because it is securely stored by the browser on the client machine. Since the WSI cannot send client certificates to the application server, the WSI will not work with an application server configured to require client certificates for authentication.
- The WSI module **must** be installed on the **same** machine as the Web server.
- For **clustered application servers**, the WSI **requires** a third-party hardware dispatcher to securely redirect requests to application servers through one port in the firewall.

  Since the application server's software Dispatcher (SilverDispatcher) load-balances server clusters by redirecting URLs (instead of masking them), the WSI modules supplied with the application server do not work with SilverDispatcher.
- For IIS and iPlanet:
  - The WSI configuration file (AgWSI.conf) **must** be in the same directory as the WSI module, since the WSI reads the configuration file when the Web server starts.
  - To run WSI modules and maintain separate log files for both IIS and iPlanet, you should create separate directories to store the WSI DLL and the shared library file with an associated AgWSI.conf file.
  - If you are **not** running both IIS and iPlanet WSI modules on the same machine, you do not need to create separate directories.

You can run the application server remotely—you do not need to run the application server from the Web server machine.

Customizing your WSI configuration

The download of the WSI modules from the Novell developer Web site includes readme files that describe how to create a basic WSI configuration.

For NetWare users For NetWare users, the Apache WSI is installed, configured, and enabled using a set of default directives.

This section describes how you can customize the default configurations by adding other directives to the WSI configuration files. It includes information on:

- Customizing the Apache WSI on NetWare
- Directing requests to multiple application servers
- Connection pooling

Customizing the Apache WSI on NetWare

To configure and enable the Apache WSI module, you add a set of directives to the Apache server's httpd.conf file, or add the directives to a separate configuration file that the httpd.conf file includes. The directives you choose determine how the WSI functions: what URLs it processes; what application server (and port) processes the URL; and whether the WSI uses connection pools, SSL, and so on.

For NetWare users If NetWare users want to use the installed settings, no action is needed. For information about other directives, see Appendix D, "WSI Directives Reference".

➤ **To configure the Apache WSI:**

1 Open the **httpd.conf** file.

2 Add an Apache **LocationMatch** directive, specifying the **URL** you want managed by the WSI. It should look something like this:

```
<LocationMatch /Root_URL_to_Forward>
```

where /*Root_URL_to_Forward* is the URL you want the WSI to forward to the application server.

To forward **all** URLs to the application server, specify:

```
<LocationMatch />
```

3 Within the LocationMatch element (on the next line), add the Apache **SetHandler** directive and specify **wsi-handler**—for example:

```
<LocationMatch /myURL>
   SetHandler wsi-handler
```

4 Specify the WSI's type of connection to the application server—pooled or nonpooled.

📖 For more information, see "Connection pooling for the Apache WSI" on page 115.

5 Save the httpd.conf file.

Directing requests to multiple application servers

The WSI module allows you to direct different client requests to different application servers:

- For Apache Web servers
- For IIS and iPlanet Web servers

For Apache Web servers

To direct different client requests to different application servers for the Apache WSI, you define multiple LocationMatch sections. Each LocationMatch section contains the directives that define the requests and the application server to which they are redirected. For example:

```
<LocationMatch URL1>
   WSIHost     host1
   WSIPort     port1
</LocationMatch>
<LocationMatch URL2>
   WSIHost     host2
   WSIPort     port2
</LocationMatch>
```

Here all URLs beginning with URL1 will be directed to host1:port1, and all URLs beginning with URL2 will be directed to host2:port2.

For IIS and iPlanet Web servers

To direct different client requests to different application servers for the IIS or iPlanet WSI, you define multiple sections in an AgWSI.conf file. Each configuration section is labeled with a **SECTION** statement and contains the statements that specify which requests are redirected to which application server. Each section must contain all statements listed as required in “AgWSI.conf file reference” on page 258.

In addition, your Web server might be *multihoming*—hosting more than one host name. Using the **WSI.host** statement, you can configure the WSI to forward requests based on the request’s host name (and optionally port). If a section does not have a WSI.host statement, the request host header is ignored and only URL matching is used as a filter. For more information, see “WSIHost” on page 254.

For each incoming request to the Web server, the WSI searches all the configuration sections until it finds a match based on the host name and port in the request header (if specified by WSI.host) and the URL of the request. The request is then forwarded to the application server specified in that section.

Sample configuration file directing to different application servers The following shows a configuration file with three sections that specify two different Web server host names and direct requests to two different application servers. The sample also shows how to configure the WSI so that specified URLs can be accessed only through a secure (HTTPS) connection:

```
SECTION=WWW_ABC_COM

# Redirect all URLs for www.abc.com to serv1.myco.com
# HTTP requests will be forwarded to port 80 on the application server
# HTTPS requests will be forwarded to port 443 on the application server

    WSI.host=www.abc.com
    WSI.root.dir=/AgISAPI

    SilverServer.host=serv1.myco.com
    SilverServer.http.port=80
    SilverServer.https.port=443
    SilverServer.urls=/

    Connection.http.max=100
    Connection.https.max=100
    Connection.idle.time=25

SECTION=WWW_XYZ_COM_SECURE

# Redirect URLs starting with /db1/approot/secure
# for www.xyz.com to serv2.myco.com
# only from HTTPS (secure port) (SilverServer.http.port is set to 0)

    WSI.host=www.xyz.com
    WSI.root.dir=/AgISAPI
```

```
        SilverServer.host=serv2.myco.com
        SilverServer.http.port=0
        SilverServer.https.port=443
        SilverServer.urls=/db1/approot/secure

        Connection.http.max=100
        Connection.https.max=100
        Connection.idle.time=25

    SECTION=WWW_XYZ_COM_HTTP

    # Redirect all other URLs for www.xyz.com
    # to serv2.myco.com on any port

        WSI.host=www.xyz.com
        WSI.root.dir=/AgISAPI

        SilverServer.host=serv2.myco.com
        SilverServer.http.port=80
        SilverServer.https.port=443
        SilverServer.urls=/

        Connection.http.max=100
        Connection.https.max=100
        Connection.idle.time=25
```

Connection pooling

The WSI modules use connection pooling to improve response time. Instead of creating and maintaining a connection to the application server for each client connected to the Web server, the WSI reuses its connections to the application server for multiple client connections. The WSI will open new connections to the application server as needed for concurrent request processing. This section includes these topics:

- Connection pooling for the Apache WSI
- Connection pooling for the IIS and iPlanet WSI

Connection pooling for the Apache WSI

In a basic Apache WSI configuration using nonpooled connections, the WSI creates a new connection for each request, runs the request, then disconnects from the server. You'll specify the application server host (WSIHost) and the application server port directives (WSIPort for HTTP requests or WSISslPort for HTTPS requests). A nonpooled connection LocationMatch directive would look like this:

```
<LocationMatch /myApp>
   SetHandler   wsi-handler
   WSIHost      alaska
   WSIPort      10080
   WSISslPort   10043
</LocationMatch>
```

When you define a connection pool, the WSI gets a connection from the connection pool for each request, runs the request, then returns the connection to the pool. The WSI saves the overhead of creating and closing connections.

To define a connection pool, you specify the connection pool attributes using a WSIConnPool container element—then within the LocationMatch directive, you specify the name of the connection pool (WSIConnPool).

First you need to create a WSIConnPool container (directly before the corresponding LocationMatch container). Here's an example:

```
<WSIConnPool cp01>
   WSIHost        alaska
   WSIPort        10080
   WSISslPort     10443
   WSIMaxConns    100
   WSIMaxSslConn  50
</WSIConnPool>

<LocationMatch /SilverStream40>
   SetHandler        wsi-handler
   WSIConnPool   cp01
</LocationMatch>
```

You can specify an idle timeout for the connections in the connection pool using the WSIIdleTimeout directive. If not specified, the idle timeout is set to 10 minutes. The WSI checks for idle connections at intervals specified by the WSICleanupInterval directive. These directives are per connection pool and have default values if not specified.

Connection pooling for the IIS and iPlanet WSI

The IIS and iPlanet WSI modules maintain separate connection pools for HTTP and HTTPS protocols. To ensure that these connection pools do not use up too many system resources, the WSI periodically scans the pools and closes any connection that has not been used between scan cycles.

Connections have the following states:

State	Description
Connected	Connection is busy and active
Inactive	Connected but time limit has not expired
Idle	Not active and idle limit has expired
Not connected	—

The WSI periodically runs a background thread to check the connection pools for inactive and idle connections to see which connections are connected but not busy. The WSI thread marks all inactive threads as idle and closes any connections that are already marked as idle. When a connection is marked as idle, that connection is closed based on the connection idle time limit you set (or the default of 25 minutes). The interval starts when the WSI module is loaded, not when the connection was created.

A connection needs to remain inactive for two scan cycles before it is disconnected. For example, if the specified connection idle time limit is set to 15 minutes, an inactive connection will be disconnected sometime between 15 and 30 minutes after becoming inactive (because the WSI may not start pooling until the middle of an existing interval). If an idle connection is used during the time interval between the scan cycles, the connection goes back into the connection pool marked as inactive.

Specifying connection pooling values You can specify the connection pooling values using three directives in the AgWSI.conf file: **Connection.http.max**, **Connection.https.max**, and **Connection.idle.time**.

Security considerations for IIS and iPlanet

Because of variations in server platforms, architecture, and third-party security providers, you may need to be aware of certain security considerations when using the WSI:

- Using IIS NTLM authentication with the WSI module
- Using the AgWSIUser utility

Using IIS NTLM authentication with the WSI module

If you secure your Web site using the Microsoft Windows NT LAN Manager (NTLM) authentication, the WSI default header settings will not work. After authenticating incoming requests, IIS adds an NTLM HTTP authentication header to each request. Because the NTLM HTTP authentication header is not supported by the application server, incoming requests will be rejected unless you configure the WSI module for IIS in one of the following ways:

- Replace all authentication headers with the ones set by the AgWSIUser utility.
- Remove the NTLM HTTP authentication headers from all requests sent to the application server. Setting **WSI.auth.NTLM.remove** in the AgWSI.conf file to **true** allows users' requests to be successfully forwarded to the application server once NTLM headers are removed. For more information, see "WSI.auth.NTLM.remove" on page 260.

Using the AgWSIUser utility

AgWSIUser is a command-line utility that generates the **WSI.auth.user** statement that is needed in AgWSI.conf to define a WSI user and password. The AgWSIUser utility encrypts the user name and password in a form the WSI can read.

📖 For more information, see "WSI.auth.user" on page 261.

➢ **To use the AgWSIUser utility:**

1. From the command line, change to your WSI root directory.
2. Enter the following command:

   ```
   AgWSIUser username password
   ```

   If your password is blank, just enter the user name. You can enter any valid user. The AgWSIUser utility prints the corresponding WSI.auth.user statement in the command window.
3. Paste the generated statement into the appropriate section of the AgWSI.conf file. (If you are not using sections, paste it anywhere in the file.)

   At startup, the WSI module will decrypt the user name and password and generate an HTTP authentication header that it will add to every request it forwards to the application server.

9 Setting Up Security

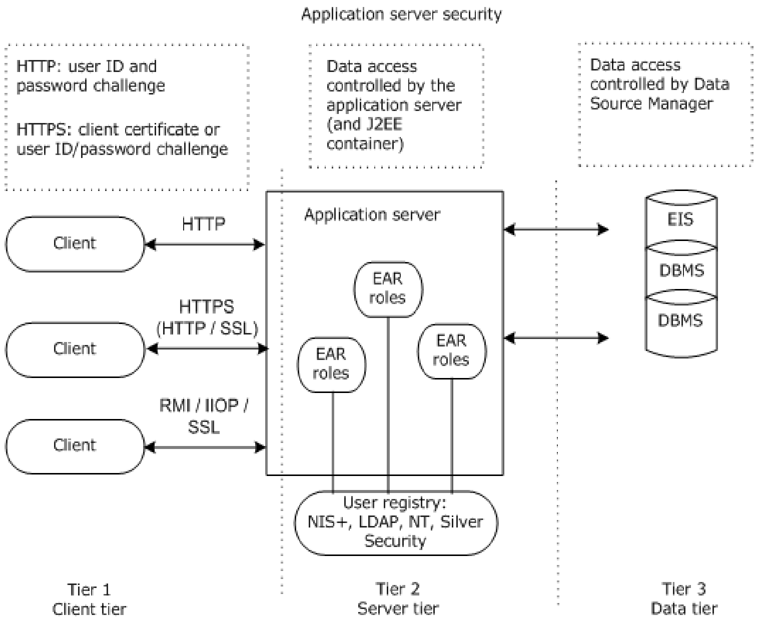
This chapter describes how to configure security for a Novell exteNd Application Server. It contains sections on:

- Security configuration
- About authentication
- Establishing a secure connection to the server
- Accessing security provider systems
- Using security provider login formats
- Using certificates
- Enabling authentication
- Using Cryptographic Hardware Integration
- Managing trusted clients
- Configuring FIPS-compliant mode

📖 For information about setting up Silver Security users and groups, see Chapter 6, “Setting Up Users and Groups”.

Security configuration

In the application server’s three-tiered architecture, security is set up at the server tier:



Application server security
HTTP: user ID and password challenge
HTTPS: client certificate or user ID/password challenge
Data access controlled by the application server (and J2EE container)
Data access controlled by Data Source Manager
Application server
Client
HTTP
Client
HTTPS (HTTP / SSL)
Client
RMI / IIOP / SSL
EAR roles
EAR roles
EAR roles
EIS
DBMS
DBMS
User registry: NIS+, LDAP, NT, Silver Security
Tier 1 Client tier
Tier 2 Server tier
Tier 3 Data tier

The application server acts as a single user with multiple connections to a database (to which applications are deployed) or a connection pool (from which data is retrieved). Acting in this manner, the application server adds extra user and object security for the data source. In effect, the application server is treated as a user with access privileges. Native data source activity and security measures are not compromised.

The application server supports HTTP, HTTP using the SSL 3.0 protocol or the TLS 1.0 Transport Layer Security protocol (HTTPS). HTTPS provides data encryption between clients and the application server to ensure privacy and data integrity.

EJBs running on the application server use the IIOP over SSL protocol to ensure privacy and data integrity. The IIOP over SSL support is provided by the ORB. The application server supports only RSA authentication for EJBs.

For more information on the ORB’s IIOP over SSL support, see the Novell exteNd Messaging Platform help.

About SSL SSL 3.0 (developed by Netscape) provides security and privacy over the Internet. TLS 1.0 (Transport Layer Security protocol) based on SSL 3.0 (defined by Internet Engineering Task Force (IEFT)) will eventually supersede SSL 3.0. HTTPS provides data encryption between clients and servers to ensure privacy and data integrity. The SSL protocol is application-independent, allowing protocols like HTTP and FTP to be layered on top of it. The SSL protocol is able to negotiate encryption keys as well as authenticate the server before data is exchanged by the higher-level application. The SSL protocol maintains the security and integrity of the transmission channel by using encryption, authentication, and message authentication codes.

You may choose to require SSL to access your application. Because SSL affects performance, you may decide to use SSL only for specific data-sensitive portions of your site. Or you could consider using Cryptographic Hardware Integration (CHI), which enables significant application server SSL encryption/decryption performance enhancements. For information on installing and using CHI on your application server, see “Using Cryptographic Hardware Integration” on page 161.

About HTTPS With HTTPS, you get a communications channel that provides privacy, user authentication, and message integrity. The application server implements SSL and TLS as follows:

- The application server must have an authentication certificate (also called public-key certificate, digital ID, or digital certificate) to make an SSL or TLS connection. The certificate is a digital “ID card” that cannot be forged. This certificate describes the server and includes a chain of trust.
- Both client and server encrypt what they send using information from both their own certificate and the certificate at the other end (if the client has a certificate). This means the sender can be sure that only the intended recipient can decrypt the data, and the recipient can be sure that the data came from the place it claims to have come from and that no tampering has occurred.

Mapping J2EE roles to existing users You can map roles defined in a J2EE archive’s deployment plan to users and groups available to the application server (Silver Security users, or users and groups from an external security system). For information on how this works, see “Accessing security provider systems” on page 124.

Types of encryption used for authentication

Authentication begins and ends with the client session. Both the RSA (Rivest-Shamir-Adleman) and DSA (Diffie-Hellman) encryption algorithms are based on public and private keys. The SSL and TLS protocols require the server to have an X.509 certificate containing its identity, its public key, and the identity and signature of the Certificate Authority (CA) that issued the certificate. The client authenticates the server based on the certificate it receives. The client then encrypts the public key and sends it back to the server to be used for encrypting further data transmissions. The encryption algorithms usually used are RC4, DES, and 3DES.

The client session operates in one of three modes:

Mode	Description
RSA (Rivest-Shamir-Adleman)	Encryption provides secure communications between Java clients, HTML clients, and the application server.
DSA (Diffie-Hellman)	Encryption provides a secure channel for the application server to communicate with Java clients.
Base64 encoding	Encryption method used with HTTP protocol (when SSL or TLS is not implemented) to send user name and password information from the client to the server. This encryption method can be decrypted easily. To ensure a secure exchange of user name and password, use an SSL or TLS connection. The SilverMaster stores Silver Security user names with encrypted passwords, or you can use any of the supported external security providers. For external security systems, the application server verifies the password information with the external security system provider.

Security functions

The application server security system provides four major security functions:

Function	Description
Authentication	This is done through a challenge, such as requesting a user ID/password pair when using HTTP, or through an authentication certificate when using HTTPS.
Access control	Once the application server verifies a user's identity, it checks whether the user is allowed to perform the requested operation on the requested object.
Data integrity	The application server ensures that the data received over the network is the same data that was sent.
Data privacy	The application server prevents unauthorized users from seeing data during transmission.

The application server security system handles all data integrity and data privacy functions with virtually no administrator involvement.

The remainder of this chapter describes how to implement authentication in the application server environment.

About authentication

Authentication is the process of determining user identity. Some applications identify users through a challenge, such as requesting a user ID/password pair, or through an authentication certificate. When an Anonymous user tries to access an object on your site, you can require a login or return an error. If you require a login, you can do so at either the server level or the object level. Use the SMC to configure specific objects to require a login for access. Alternatively, use the **Require user authentication** setting at the server level to force users to log in when they first connect to the server.

For more information, see "Enabling authentication" on page 159.

Establishing a secure connection to the server

If you intend to use SSL or TLS communications, you need to:

- Obtain a server certificate and install it on the server
- Enable the RSA and/or DSA ports
- Disable HTTP (if you want to require only SSL)
- Enable TLS 1.0 and/or SSL 3.0

When doing administrative tasks such as adding users and databases to the server, you may want a secure (SSL) connection between the server and the client you are using (such as the SMC or a browser) so that all communication is encrypted.

The following three sections describe:

- Establishing a secure (SSL) connection between a Java client and the application server
- Establishing a secure (SSL) connection between an HTML client and the application server
- Establishing a secure (SSL) connection between an EJB client and the application server

Establishing a secure (SSL) connection between a Java client and the application server

Secure communications between the application server and Java clients can use either the RSA or the DSA protocol.

Because you can configure unique ports for each protocol, the port you specify depends on whether it is intended to be a runtime port for users or an administration port.

For more information, see "Setting up separate ports" on page 79.

➤ To establish a secure (SSL) connection between a Java client and the application server:

1. Install an RSA or DSA certificate on the application server.

   For information, see "Using certificates" on page 136.

2. Enable the RSA or DSA port(s) in the SMC.

   For information, see "Enabling RSA/DSA ports" on page 149.

3. Connect your client to the server using HTTPS at the DSA or RSA port.
4. In the dialog that displays, specify your server followed by the number of the runtime or administration port you want to use.

   - For an RSA port, specify **https://*server:RSA_port***.

     For example:

     ```
     https://tara:port
     ```

   - For a DSA port, specify **https://*server:DSA_port*** on the command line for the hostname.

     For example:

     ```
     https://tara:443
     ```

NOTE: If you want to use the RSA port default, you need only specify **https://*hostname*** on the command line. If you want an RSA connection on a port number other than 443, you must specify the port value on the command line.

Establishing a secure (SSL) connection between an HTML client and the application server

Secure communications between the application server and an HTML client (browser) use the RSA protocol.

➢ To establish a secure (SSL) connection between an HTML client and the application server:

1 Install an RSA certificate on the application server.

   📖 For information, see "Using certificates" on page 136.

2 Enable the RSA port in the SMC.

   📖 For information, see "Enabling RSA/DSA ports" on page 149.

3 Open your browser to the server using HTTPS at the RSA port.

   The RSA port you specify depends on what type of operations you want to perform.

   - To run the application, specify the runtime port.
   - If you use a custom HTML administration tool, specify the administration port.

   Specify your server followed by the (optional) number of the RSA runtime or administration port:

   ```
   https://server[:port]
   ```

   For example:

   ```
   https://tara:443
   ```

NOTE: If you want to use the RSA port default, you need only specify **https://*hostname*** on the command line. If you want an RSA connection on a port number other than the 443, you must specify the port value on the command line.

Establishing a secure (SSL) connection between an EJB client and the application server

Secure communication between the application server and EJB clients is established using the IIOP over SSL capabilities of the ORB. At startup, the application server exports the RSA certificate to the ORB. When the deployment plan of an EJB specifies a cipher suite and the application server has an RSA certificate installed, the ORB ensures that the communication is secure.

Java clients require access to the agrootca.jar file in order to participate in secure communications. This file is installed in the **\Common\lib** directory. This file is installed automatically for SilverJ2EEClient clients.

Communications failures Communications failures may happen when:

- The server does not have an RSA certificate installed
- SilverJ2EEClient does not have the matching CA certificate of the server's certificate
- SilverJ2EEClient does not have the agrootca.jar file installed

📖 For information on specifying cipher suites for deployed EJBs, see the chapter on J2EE archive deployment in the *Facilities Guide*.

➢ To establish a secure (SSL) connection between an EJB client and the application server:

1 Install an RSA certificate on the application server.

   📖 For information, see "Using certificates" on page 136.

2 Enable the RSA runtime port in the SMC.

   📖 For information, see "Enabling RSA/DSA ports" on page 149.

3 For HTML or Java clients, connect your browser to the server using HTTPS at the RSA runtime port:

   ```
   https://server:RSA_port_rt
   ```

For example:

```
https://tara:443
```

OR

```
https://tara
```

NOTE: If your RSA runtime port uses port 443 (the default), you need only specify **https://*hostname*** on the command line. If you want an RSA connection on a port number other than 443, you must specify the port value.

For EJB applications that contain stateful session beans for which session-level failover is specified, you must also create a range of ports for IIOP SSL communications.

📖 For more information on creating the IIOP SSL port range, see "Specifying ORB settings" on page 84.

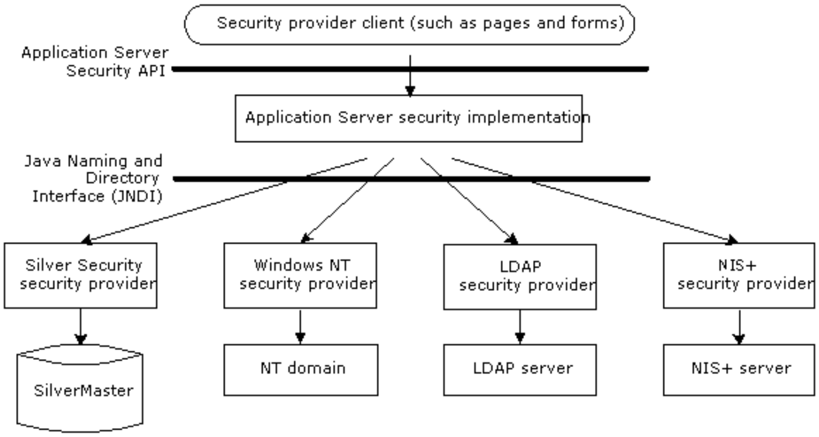
Accessing security provider systems

The system verifies users and their permission levels according to lists of groups and users that you provide. User and group information can be defined using the application server's native security system called Silver Security, or it can be obtained from an external security system. All Silver Security information is stored in the SilverMaster database. For external security, all information is obtained from the external system.

The application server recognizes users and groups from the following systems:

Security provider	Description
Silver Security	Native security that maintains a list of valid users and groups in the SilverMaster database.
Windows NT directory services	Capability to connect the application server to the NT Domain name registry.
Lightweight Directory Access Protocol (LDAP)	Directory service that connects the application server to defined LDAP directories.
NIS+	Network Information Services Plus, a name service that is available on SunOS 5.x and later operating systems.
X.509 certificates	The application server supports client certificates generated from authority servers such as VeriSign, Netscape Certificate Server, and Microsoft Certificate Server. For more information, see "Using certificates" on page 136.

The application server implements the Java Naming and Directory Interface (JNDI), which connects it to native UNIX security and to Windows NT and LDAP directories:


Security provider client (such as pages and forms)
Application Server Security API
Application Server security implementation
Java Naming and Directory Interface (JNDI)
Silver Security security provider
Windows NT security provider
LDAP security provider
NIS+ security provider
SilverMaster
NT domain
LDAP server
NIS+ server

Adding security provider access

You can use the SMC to set up access to security provider systems. After you set up access to provider directories, you can define access control for the users and groups from these external systems.

If you want to change from using Silver Security to a new security provider, make sure your administrator account has access permissions to the new security provider account.

CAUTION: *If you disable Silver Security before you grant the administrator account access to the new security provider, you will need to run SilverMasterInit -l to regain access to the application server.*

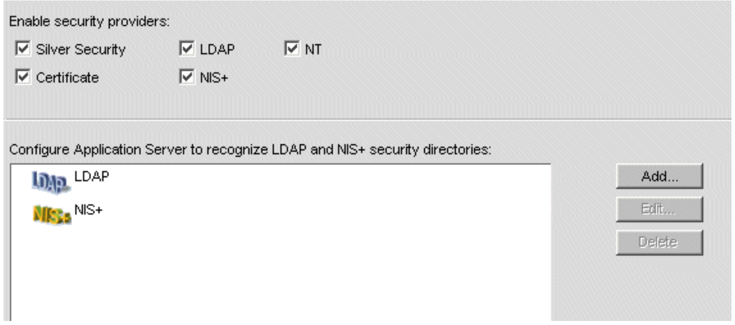
For information about setting up Silver Security users and groups, see Chapter 6, "Setting Up Users and Groups". For information about access control, see "Authorization and access control" on page 165.

To add security provider access:

1 Start the SMC.

2 Select the **Security** icon on the toolbar.

3 Select **Security Providers**:



Enable security providers:

☑ Silver Security ☑ LDAP ☑ NT

☑ Certificate ☑ NIS+

Configure Application Server to recognize LDAP and NIS+ security directories:

- LDAP
- NIS+

Any LDAP and NIS+ servers that are known to the application server are listed.

4 Select the type of provider you want to register (all providers are selected by default).

- **NT** is valid only when running **Windows NT**. If you choose NT, it is not necessary to add an NT domain; NT provides system calls that the application server uses to discover the primary and trusted domains. But you do need to set up the server so that users can log in with their NT names. See "Using NT security" on page 127.
- **NIS+** is valid only when you are running **Solaris**.

5 To add an LDAP or NIS+ server connection, choose the appropriate item and click **Add**.

📖 For more information on adding the LDAP security provider, see "Using LDAP security" on page 127.

📖 For more information on adding the NIS+ security provider, see "Using NIS+ security" on page 132.

Resetting the security resource timeout

The SMC also allows you to set the security resource timeout period,which defines how often the application server reloads the list of available users and groups from defined security providers. The default value is 15 minutes.

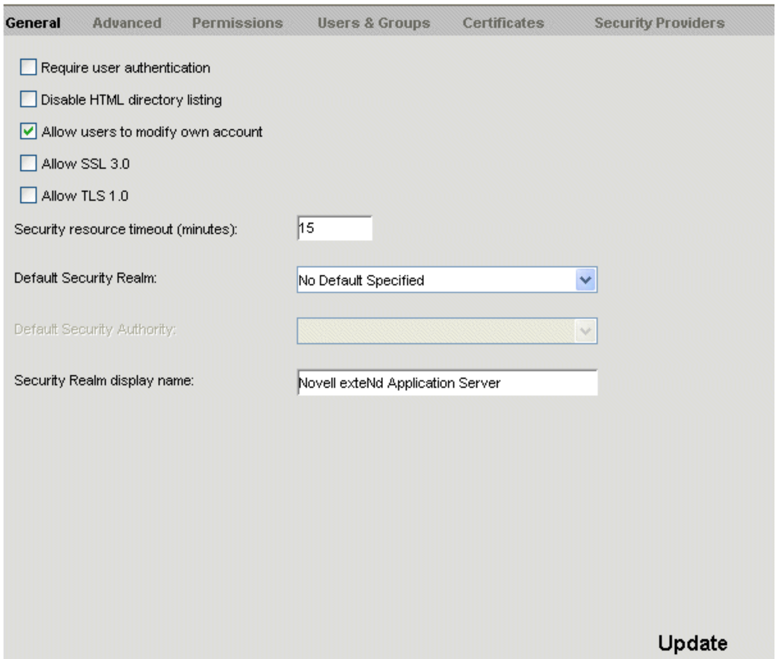
You may want to increase this number if the information in the external system does not change frequently or if the connection to it is slow.

➢ To reset security resource timeout:

1 Start the SMC.

2 Select the **Security** icon from the toolbar.

3 Select **General**:



☐ Require user authentication

☐ Disable HTML directory listing

☑ Allow users to modify own account

☐ Allow SSL 3.0

☐ Allow TLS 1.0

Security resource timeout (minutes): 15

Default Security Realm: No Default Specified

Default Security Authority:

Security Realm display name: Novell exteNd Application Server

4 Edit the **Security resource timeout** value as needed.

Using NT security

You can use various NT directory services to manage NT users and groups. For example, users logging in with their NT user name and password only have to do this once per session (unless you set additional security at the server or cluster level).

Local and global groups

Using NT users and groups can help simplify security administration. For example, defining local groups lets you combine users and global groups from multiple domains into a single group.

A **local group** is available only within the domain in which it is created. A **global group** is available within its own domain as well as any trusted domain.

Local groups defined on the server machine can include local users as well as global groups and users from the primary domain or any trusted domain. An NT local group cannot, however, contain other local groups.

For more information on NT user groups, see your Windows NT documentation.

Speeding NT authentication Supporting local groups can result in slower NT authentication if you have many large trusted domains. If this is a problem for you, you can speed authentication by disabling local group support — in either of two ways:

- Adding the following line to the httpd.props file (located in the application server's **\Resources** directory):

  ```
  http-server.com.sssw.srv.SupportNTLocalGroups=false
  ```

- Programmatically setting the property PROP_SUPPORT_NT_LOCAL_GROUPS (in AgiAdmServer and AgiAdmCluster) to Boolean.FALSE

NT privileges needed when running the server as a service

To support NT users and groups, the application server requires the **Act as Part of the Operating System** and **Log on as a Service** operating system privileges. These privileges are set when the application server is configured as a service running under the default NT System account. However, if you change the service to run under a user account or if you decide to stop running the server as a service, you must make sure that these two NT system privileges are set.

If you change the service to run under a user account, the NT Control Panel automatically grants the **Log on as a Service** privilege to that account; however, you will need to manually configure the **Act as Part of the Operating System** privilege. To allow users from trusted domains to log in, you must configure the **Log on Locally** privilege regardless of whether or not the application server is running as a service.

For more information on setting up Windows security, see your Windows documentation.

Using LDAP security

Lightweight Directory Access Protocol (LDAP) is a directory service that allows Internet clients to query and manage an arbitrary database of hierarchical attribute/value pairs over a TCP/IP connection. LDAP provides a specification that allows applications to communicate with it. The application server allows you to specify LDAP users and groups, display LDAP attributes, and use LDAP users and groups in access control expressions. The application server supports access to LDAP servers (such as Novell eDirectory™, Microsoft Active Directory and Sun One Directory Server) that support both Version 2 and Version 3 of the LDAP protocol.

Access to LDAP information

The application server interacts with an LDAP server as follows:

When the application server interacts with an LDAP server	Details
When it needs to verify a user's credentials during login	In this case, the application server passes a **specific user's** login information to the LDAP server.
When it needs to display **generic** information such as lists of users and groups	How (or whether) your application server accesses this generic information depends on how your LDAP server is configured: ◆ If your LDAP server **does not allow anonymous access**, you can configure the application server to pass the system login credentials to the LDAP server. You need to provide these system credentials in Step 5 of the setup procedure that follows. ◆ If your LDAP server **allows anonymous access**, no system login credentials need to be passed. You do not need to specify a system account for LDAP in Step 5.

Connecting to LDAP servers using SSL

To prevent information about LDAP groups and users (including client credentials) from being transferred as clear text, use an SSL or TLS connection between the application server and the LDAP server.

To use SSL or TLS communications with the application server, you must already have configured your LDAP server to support SSL or TLS and have installed certificates on the LDAP server.

📖 For more information, see your LDAP server documentation.

NOTE: When using SSL or TLS communications with LDAP, you can set the application server certificate to be sent to the LDAP server if it's requested. When the LDAP server is set to request or require certificates, it will attempt to verify any certificates sent to it.

Connecting to LDAP servers that support only LDAP Version 2

You can set the application server connection to use only the LDAP Version 2 protocol. By default, the application server first tries to connect to the LDAP server using LDAP Version 3. If the connection attempt fails, the LDAP server is supposed to report an error, in which case the application server will try to connect using the Version 2 protocol.

If LDAP Version 3 is not supported This approach will not work with an LDAP server (such as Microsoft Site Server) that doesn't always report the error using LDAP Version 3. If LDAP Version 3 is not supported, you need to set the **Force LDAP Version 2** option in the SMC in Step 6 in the setup procedure.

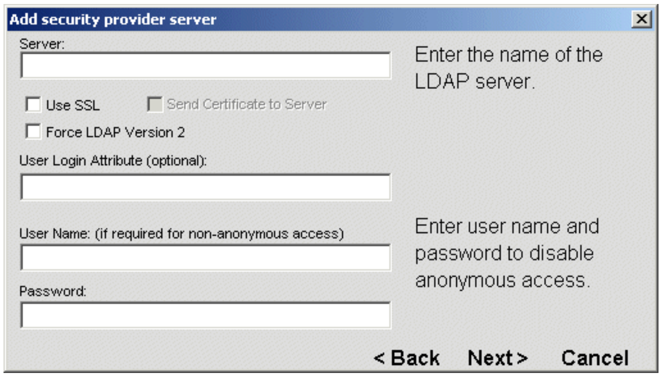
➢ **To set up LDAP security:**

1. Start the SMC.
2. Select the **Security** icon from the toolbar.
3. Select **Security Providers**.
4. Select **LDAP** in the provider list, then click **Add**.

   A wizard displays.

5 Select **LDAP** and click **Next**.

The following panel displays:



Add security provider server

Server:

Enter the name of the LDAP server.

☐ Use SSL ☐ Send Certificate to Server

☐ Force LDAP Version 2

User Login Attribute (optional):

User Name: (if required for non-anonymous access)

Enter user name and password to disable anonymous access.

Password:

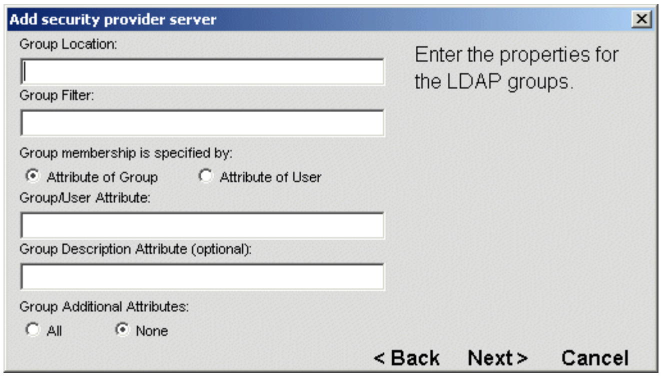
6 Specify the server and (optionally) the login attribute and user name/password as follows.

Field	What you specify
Server	The name of the LDAP server. The server name must be recognized on your network. If the LDAP server uses a nondefault port, you must specify it as part of the server name. For example: **localhost:636**
Use SSL	Use this option if the specified LDAP server and port are configured to use SSL communications. 📖 For more information, see “Connecting to LDAP servers using SSL” on page 128.
Send Certificate to Server	This option allows the application server’s certificate to be sent to the LDAP server. When the LDAP server is set to request or require certificates, it will attempt to verify any certificates sent to it. If the LDAP server is set to request or require certificates (and this option is enabled), the application server’s certificate is sent to the LDAP server so that it can verify it against a list of trusted CA certificates. If the LDAP server is set to request or require a certificate (and the application server does not have one), the application server will ignore the **Send Certificate to Server** command for certificate **requests**—and if the LDAP server **requires** a certificate, the connection will fail. You can select the **Send Certificate to Server** check box onlyif you have also selected **Use SSL** (above).
Force LDAP Version 2	Set this option to work with an LDAP server (such as Microsoft Site Server) that does not support LDAP Version 3. 📖 For more information, see “If LDAP Version 3 is not supported” on page 128.
User Login Attribute	(Optional) If you specify a value for this property, it defines the LDAP attribute that can be used to uniquely identify a user. Make sure you pick an attribute that is unique for all users. TIP: Specifying a value here can simplify login for LDAP users. For more information, see “Simplifying login for LDAP users” on page 135.

Field	What you specify
User Name and Password	If appropriate, enter a user name and password to allow the application server access to LDAP information. The application server will use these system login credentials anytime it needs to access generic LDAP server information. If your LDAP server **allows** anonymous access, these account values are **not required**. 📖 For more information, see “Access to LDAP information” on page 128.

7 Click **Next**.

The following panel displays:



Add security provider server

Group Location:

Group Filter:

Group membership is specified by:

● Attribute of Group ○ Attribute of User

Group/User Attribute:

Group Description Attribute (optional):

Group Additional Attributes:

○ All ● None

Enter the properties for the LDAP groups.

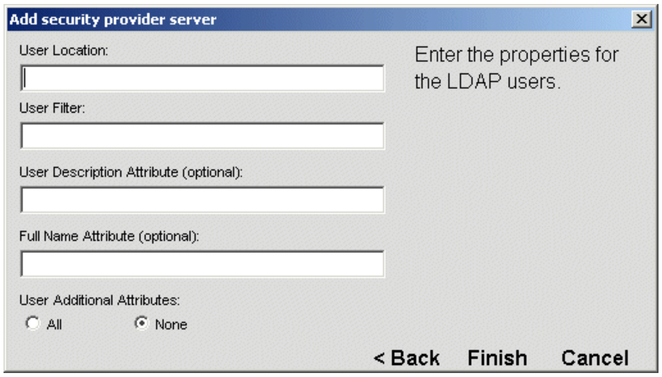
Use this panel to specify groups on the LDAP server:

Item	Description
Group Location	(Required) A distinguished name that identifies the level in the hierarchy where you want to start searching for group entries. For example, to start at an organizational unit called **employees** that exists under the organization called **myco**, enter the following: `ou=employees,o=myco` Every group under and including **employees** in the hierarchy will be included. 📖 For more information about distinguished names, see “Simplifying login for LDAP users” on page 135.
Group Filter	(Required) The LDAP search filter is used to determine what constitutes a group for this LDAP server. A common usage is to specify a value of the object class attribute that identifies a group. The filter definition can be any valid LDAP search filter. For example: `(objectclass=groupofuniquenames)`
Attribute of Group	Required for LDAP servers (like Netscape Directory Server) that use an attribute of a **group** object to define **group** membership.
Attribute of User	Required for LDAP servers (like Microsoft Site Server) that use an attribute of a **user** object to define **group** membership. NOTE: Microsoft’s Active Directory supports both Attribute of Group and Attribute of User to define group membership. Specifying Attribute of User and a User/Group Attribute of **memberOf** is the most efficient setting.

Item	Description
Group/Users Attribute	(Required) An attribute used to display all members (users) of a group in the SMC. The name you enter is the LDAP group or user attribute that defines group membership. For example: `uniquemember`
Group Description Attribute	(Optional) An attribute used to identify a group description in the SMC. The name you enter is the LDAP attribute to which you want to map the description. For example: `notes`
Group Additional Attributes	Select **All** if you want all of the specified LDAP group attributes to be listed in the SMC. Select **None** if you want no additional attributes to appear. The specified attributes will be displayed in a tab when you select a group in the Users & Groups panel and open the Property Inspector. 📖 For more information, see "Accessing users and groups" on page 132.

8 Click **Next** when you have finished specifying groups.

This panel asks you to specify users on the LDAP server:


Add security provider server
User Location:
User Filter:
User Description Attribute (optional):
Full Name Attribute (optional):
User Additional Attributes:
○ All
● None
< Back
Finish
Cancel

Enter the properties for the LDAP users.

9 Specify users as follows:

Item	Description
User Location	(Required) A distinguished name that identifies the point in the hierarchy where you want to start searching for users. For example, to start at a point (or node) entitled **developers** that exists under **software**, enter the following: `ou=developers,o=software` Every user under and including **developers** in the hierarchy will be included.
User Filter	(Required) The LDAP search filter is used to determine what constitutes a user for this LDAP server. A common usage is to specify a value of the object class attribute that identifies a user. The filter definition can be any valid LDAP search filter. For example: `(objectclass=person)`
User Description Attribute	(Optional) An attribute used to identify a user description in the SMC. The name you enter is the LDAP attribute to which you want to map the description. For example: `title`
Full Name	(Optional) Specifies the full name attribute, if available. For example: `cn`

Item	Description
Additional Attributes	Select **All** if you want all of the specified LDAP user attributes to be listed in the SMC. Select **None** if you want no additional attributes to appear. The specified attributes will be displayed in a tab when you select a user in the Users & Groups panel and open the Property Inspector. 📖 For more information, see "Accessing users and groups" on page 132.

10 Click **Finish**.

The SMC displays the settings under the LDAP directory. You can view the new settings anytime by selecting **Users & Groups** in the Security options in the SMC.

Using NIS+ security

NIS+ (Network Information Services Plus) is a name service available on SunOS 5.x and higher operating systems. Users are contained in the **NIS+** table identified by passwd.org_dir, and groups by group.org_dir. After you have added users and groups, you can use them in security expressions for access control.

➢ To set up NIS+ security:

1 Start the SMC.

2 Select the **Security** icon from the toolbar.

3 Select **Security Providers**.

4 Select **NIS+** from the provider list, then click **Add**.

A wizard displays.

5 Select **NIS+** and click **Next**.

6 Type the name of the **NIS+** server in the following format:

```
servername/nisDomain.com\username
```

The server name must be recognized on your network.

Accessing users and groups

You can use the SMC to view users and groups that you have defined for any security provider.

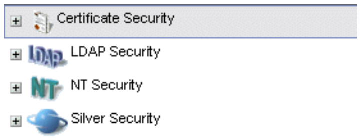
➢ To view users and groups:

1 Start the SMC.

2 Select the **Security** icon from the toolbar.

3 Select **Users & Groups**.

4 Highlight an icon to view the users and groups known to the server:

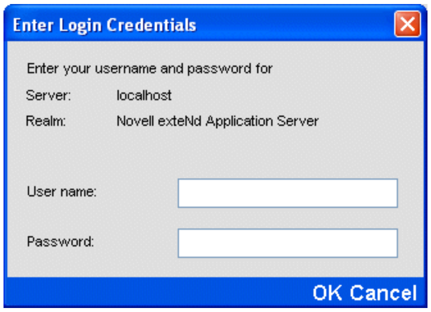


You can expand a selected item to show specific users and groups.

Using security provider login formats

The application server supports a number of security realms including Silver Security, NT security, LDAP security, NIS+ security, and certificate security. All except certificate security involve establishing an identity by providing user name information and a password.

When users log in they see a dialog similar to this:



Enter Login Credentials

Enter your username and password for

Server: localhost

Realm: Novell exteNd Application Server

If the authentication dialog is being presented by a browser, the dialog—defined by the browser—will look a little different from the dialog shown above but will consist of the same fields.

Colons cannot be used in user names or passwords

User authentication in HTTP works by taking the user name and password separated by a colon (:). So make sure that user names and passwords don't contain colons. With LDAP distinguished names in particular (they can be quite long), make sure no component of the name contains a colon.

User name parts

A user name is composed of three parts delimited by backslash characters: Realm\Authority\Name.

Part	Description
Realm	The application server supports the following security provider realms for login: ◆ SSSW (for Silver Security users) ◆ NT ◆ LDAP ◆ NIS+
Authority	Authorities are as follows: ◆ For Silver Security, there is no authority ◆ For NT, the authority is the NT domain ◆ For LDAP, the authority is the server ◆ For NIS+, the authority is the server name and the domain name, separated by a forward slash (/)
Name	The user name

User name shortcut formats By default, the application server allows shortcuts to the full user login name, as follows:

- **Silver Security** If the user enters one part for a user name, a Silver Security user name is assumed. For example:

  emilyh is translated to **SSSW\\emilyh**

NOTE: For Silver Security, the authority is an empty string—nothing between the two backslashes. The authority is not needed, because an external security system is not used.

- **Windows NT** If the user enters two parts to a user name, an NT user name is assumed, with the format ***domain\userName***. For example:

  mydomain\craigh is translated to **NT\mydomain\craigh**

By default, LDAP and NIS+ names must be fully qualified, as follows:

- **LDAP** Login syntax for user name: **LDAP\*serverName\distinguishedName***

  The user must enter the entire pathname. For example:

  ```
  LDAP\myServer\cn=Nancy Smith,ou=My Company
  ```

- **NIS+** Login syntax for user name: **NisPlus\*server/nisDomain\username***

  The user must enter the entire pathname. Note that the authority has two components, separated by a forward slash. For example:

  ```
  NisPlus\myServer/domain1.com\jeanw
  ```

You can change these default login specifications, as described next.

Overriding defaults for login name components

You can override the defaults for login name components. So if you use only one type of external security system (and perhaps only one external security authority), you can allow users to provide a shortened name, thus simplifying the login procedure.

➢ **To override the defaults for login name components:**

1. Start the SMC.
2. Select the **Security** icon from the toolbar.
3. Select **General**.
4. Specify your default realm and (optionally) your default authority, and a display name for the realm:

Field	Description
Default Security Realm	Defines the security realm for any login name that does not explicitly define a realm. Possible values are Silver Security, NT, LDAP, NIS+, or No Default Specified.
Default Security Authority	(Enabled only if a default security realm has been defined) Defines the authority for any login name that does not explicitly define an authority. The application server provides a list of valid authorities based on the selected default realm.
Security Realm display name	A String that will be displayed as the security realm in a server login dialog. This value is also passed on the WWW-Authenticate response header sent to the client.

A full login name can always be specified, in which case the defaults are ignored.

Example Suppose your site uses security names from a single LDAP server. You could set the following defaults:

Option	What you specify
Default Security Realm	LDAP
Default Security Authority	*ServerName*

Users that exist in the LDAP server can now just use their LDAP user name and password when logging in to the application server.

In this same example, a user that exists as part of the Silver Security security realm must now specify a full login name:

```
SSSW\\SilverName
```

Simplifying login for LDAP users

In LDAP, a user name is specified relative to the LDAP naming hierarchy as a *distinguished name* (DN). The DN is a comma-separated list of nodes that contain attribute/name pairs from the leaf node where the user resides back to the root node.

By default, LDAP users logging in to the application server are required to enter the entire DN, which can be quite long. You can simplify login for LDAP users by specifying the User Login Attribute property when adding an LDAP server as a security provider. For more information on this property, see "Using LDAP security" on page 127.

If you have specified a User Login Attribute, when the user credentials are being verified during the login sequence, a search is performed for the specified User Login Attribute, with a value that matches the Name portion of the login user name. The search starts from a point in the LDAP hierarchy identified as User Location when defining the LDAP server to the application server (see "Using LDAP security" on page 127).

If the search is successful, the DN of the corresponding user (the first one found if there are multiple hits) is used to construct a fully qualified login name, and the login operation continues. If the search is not successful, the operation continues as though the Name field were the distinguished name of the LDAP user. This allows LDAP logins using either form when the attribute is set.

Example 1 Assume the following server properties are specified:

Field	What you specify
Default Security Realm (specified in Server Security panel in SMC)	LDAP
User Login Attribute (specified when defining the LDAP server to application server)	mail
User Location (specified when defining the LDAP server to the application server)	o=My Company,c=US

In this example, a default security realm is defined and the login attribute is set to **mail**. At this site, each user's **mail** attribute is that user's full e-mail address.

A user defined in an LDAP server named myServer, with a DN of **uid=ecraig,ou=Development,ou=Billerica,o=My Company,c=US** and an e-mail address of **ecraig@mycompany.com** (that is, a user whose **mail** attribute is ecraig@mycompany.com), could use a login name of either:

```
myServer\uid=ecraig,ou=Development,ou=Billerica,o=My Company,c=US
```

OR

```
myServer\ecraig@mycompany.com
```

Example 2 Assume the following server properties are specified:

Field	Value
Default Security Realm (specified in Server Security panel in SMC)	LDAP
Default Security Authority (specified in Server Security panel in SMC)	myServer
User Login Attribute (specified when defining the LDAP server to the application server)	uid
User Location (specified when defining the LDAP server to the application server)	o=My Company,c=US

In this example, a default security authority has now been specified in addition to the default security realm. The login attribute has now been set to **uid**.

The same user as above (whose **uid** is **ecraig**) could use a login name of either:

```
uid=ecraig,ou=Development,ou=Billerica,o=My Company,c=US
```

OR

```
ecraig
```

Using certificates

Certificates are required when using HTTP with the SSL 3.0 and TLS 1.0 protocols (HTTPS). HTTPS provides data encryption between clients and the server to ensure privacy and data integrity. Certificates can also be used to authenticate users.

This section contains the following topics:

- About certificates
- Creating and installing server certificates using the SMC
- Creating and installing server certificates for the dispatcher
- Viewing server certificates
- Enabling RSA/DSA ports
- Turning off HTTP communications
- Allowing SSL 3.0 and TLS 1.0
- Restricting SSL cipher suites
- Managing Certificate Authorities
- Installing and managing client certificates
- Verifying SSL server certificates for Java clients

About certificates

A *certificate* (also called a public-key certificate, digital ID, or digital certificate) is a file that authenticates the identity of a user or a group. The certificate is a kind of license issued by a trusted organization called a Certificate Authority (CA). A CA may be an external company that offers certificate services (such as VeriSign) or an internal organization such as a corporate MIS department.

For Internet applications, it is generally a good idea to have a server certificate that is signed by a widely recognized and trusted guarantor. For intranet applications, it may be sufficient to have the guarantor be the company in which the application is running.

Both users and servers can have certificates attesting to their identity. If you want to use SSL or TLS for privacy, the application server **must** have a server certificate. Once it is enabled, the server may request a client certificate from the browser attesting to the identity of the user.

Advantages of certificates

Certificates provide these important security services:

Service	Description
Enhanced authentication and security	Clients can be assured that they are communicating with who they think they are. Similarly, applications can be sure of who their users are. Certificate-based authentication is more secure than traditional methods of user authentication, such as user name/password.
Real-time encryption over SSL	The SSL encryption scheme requires the server to present its digital certificate to the client as part of the SSL handshake. Verifying the server's certificate gives the client a level of trust about the server's identity.
Convenience	Certificates allow users to log in once (for example, to the browser—a local operation), and then the browser presents client certificates to servers as needed for all other logins.

Certificate support

The following table describes how the application server supports certificates:

Support item	Description
Server certificates	A server certificate is required for SSL or TLS/HTTPS. This allows clients to authenticate the server. There are two types of certificates, depending on the type of client: ◆ **RSA-encoded server certificates** are supported for HTTPS/SSL communications between the application server, Java clients, and HTML clients. ◆ **DSA server certificates** are supported for HTTPS/SSL communications between the application server and Java clients. NOTE: Java clients verify server certificates against a list of trusted CA certificates that are stored in a JAR file when establishing an SSL connection to the application server. See "Verifying SSL server certificates for Java clients" on page 158.
Client certificates	Client certificates are optional and are used for user authentication by the server. They are installed in a browser. You can get client certificates from a number of authorities, including VeriSign. Each client certificate includes the CA certificate that generated it. The server must have a corresponding CA certificate. NOTE: The application server does not support DSA client certificates.
CA certificates on the server	CA certificates represent trusted clients based on the CA that signed for them. CA certificates are required on the server for verifying corresponding HTML client certificates. The server will authenticate only client certificates that were generated and/or signed by one of its installed CA certificates.

About global certificates

What are commonly called *global certificates* are actually **Global Secure Site IDs** from VeriSign. They are a form of digital ID that allows for 128-bit encryption worldwide. (Standard VeriSign digital IDs, now called Secure Site IDs, do not allow for U.S.-based companies to use 128-bit encryption outside the U.S.)

It is up to VeriSign to certify servers as supporting Global Secure Site IDs. It is not up to a server vendor—such as Novell—to declare support for Global Secure Site IDs.

📖 For more information, see http://digitalid.verisign.com/server/global/help/globalFAQ.htm.

Creating and installing server certificates using the SMC

You can use the SMC for these tasks:

- Generating and installing RSA server certificates
- Generating and installing DSA server certificates

Generating and installing RSA server certificates

The SMC provides the following functions for RSA server certificates:

Function	Description
Generate Request	Generates a certificate signing request (CSR) for an RSA certificate. The CSR is an encrypted file that contains information that identifies the organization running the application server. The CSR is submitted to a certificate issuer (such as VeriSign) that will use the information to create a certificate signed by the issuer. Generates the public/private key combination and stores the private key information in the database. Private key information is never passed from the server to the SMC, so this is extremely secure.
Install Certificate	Prompts for the certificate information returned by the issuer and then installs the certificate (without private key information) into the application server. It does not install the private key information, because that already resides on the server from the **Generate Request** process. If the server already has an RSA certificate for the same DNS name, installing another certificate for the same DNS name will overwrite the existing one. This function is similar to running the AgDigitalIDStep2 utility.
Import/Export Certificate	**Export** allows you to export an installed certificate and private key to a specified file on the server. The certificate and private key are exported in the standard PKCS12 format. This is used for backup. You are prompted to provide a password for the exported file. **Import** allows you to import an exported certificate and private key. Import installs the certificate on the server and will overwrite an existing certificate if the same DNS names are specified. When importing a certificate that was exported, you are prompted for a password. This is the same password used on the export.
Export Private Key	Writes the private key to the specified file on the server machine in the PKCS8 format. You are prompted for a password to protect the private key. Use this to back up your private key after using the **Generate Request** feature.

Function	Description
Install with Private Key	Prompts for the certificate information returned by the issuer and the private key file (generated via the **Export Private Key** function), then installs the certificate (with the private key information) into the application server. Use this feature when you've generated a CSR and are unable to use the private key installed via the **Generate Request** feature (for example, if the server's database has become corrupted before you receive the certificate back from the CA).

NOTE: To generate a certificate for a dispatcher, you must use the command-line tool described in "Creating and installing server certificates for the dispatcher" on page 144.

➢ To generate an RSA server CSR:

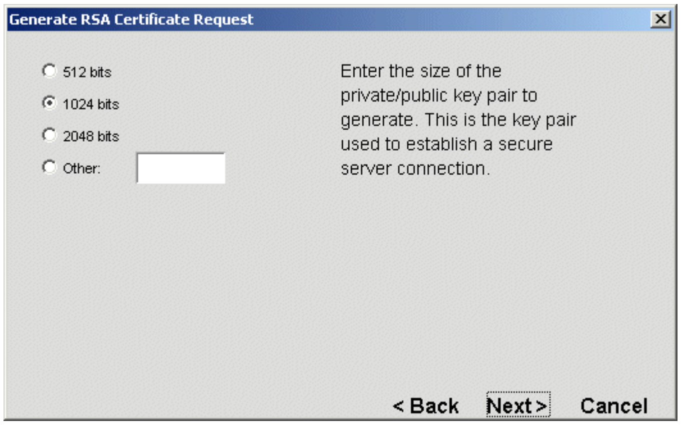
1. Start the SMC.
2. Select the **Security** icon from the toolbar.
3. Select **Certificates**.
4. Select the **RSA** tab.
5. Choose **Generate Request**.
6. Complete the items on the panel that displays as follows:

Field	What to specify
Server DNS Name	The TCP/IP hostname, which may be different from your machine name. (You can issue **ping localhost** from the command line to determine the TCP/IP name of the local host.)
Organization	Your company name in full unabbreviated legal form.
Organizational Unit	(Optional) Your department within the company.
City/Locale	(Optional) The city or locale where your company does business.
State/Region	The full name of the state or region where your company does business. Do not abbreviate.
Country	The country where your company does business. You must use the ISO two-letter country code. For example, the ISO code for the United States is US.

7. Click **Next**.

   This panel allows you to specify the size of the key pair to generate.

The 1024 bits option usually provides an acceptable level of security. Selecting a higher level decreases the speed of the initial connection. The 512 bits option provides a low level of security.


Generate RSA Certificate Request

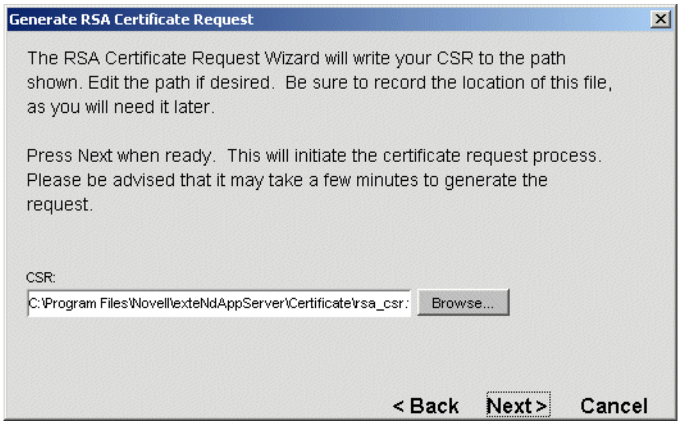
○ 512 bits
● 1024 bits
○ 2048 bits
○ Other:

Enter the size of the private/public key pair to generate. This is the key pair used to establish a secure server connection.

8 If prompted, specify the size of the key pair to generate.

9 Click **Next**.

The following panel displays:


Generate RSA Certificate Request

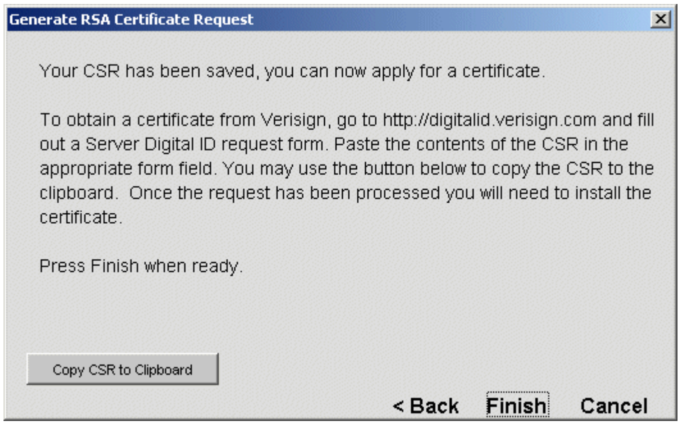
The RSA Certificate Request Wizard will write your CSR to the path shown. Edit the path if desired. Be sure to record the location of this file, as you will need it later.

Press Next when ready. This will initiate the certificate request process. Please be advised that it may take a few minutes to generate the request.

CSR: C:\Program Files\Novell\exteNdAppServer\Certificate\rsa_csr:

This panel shows the paths for the CSR. You may edit these paths if you choose. You will use this information later when installing the certificate.

10 Click **Next**:


Generate RSA Certificate Request

Your CSR has been saved, you can now apply for a certificate.

To obtain a certificate from Verisign, go to http://digitalid.verisign.com and fill out a Server Digital ID request form. Paste the contents of the CSR in the appropriate form field. You may use the button below to copy the CSR to the clipboard. Once the request has been processed you will need to install the certificate.

Press Finish when ready.

11 Click **Copy CSR to Clipboard** to copy the contents of the CSR to the clipboard for use in the next step.

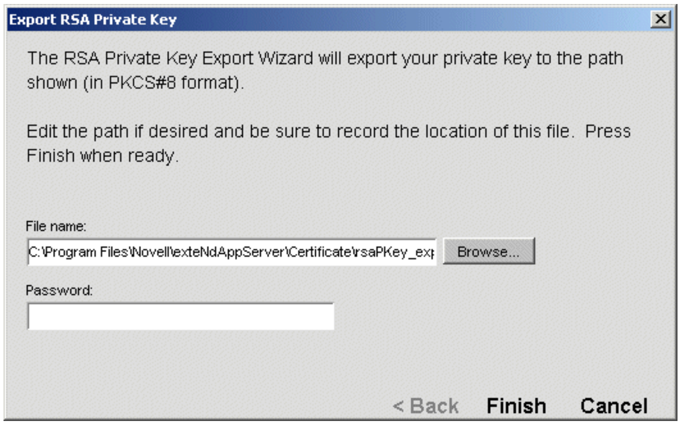
12. Follow the directions to request a certificate for your application server (for example, by using the VeriSign Web site http://digitalid.verisign.com). Once your request is approved, the certificate authority sends the new certificate back to you through e-mail.
13. Click **Finish**.

➢ To back up the private key:

After you have generated a CSR, you may want to back up the private key information in case the SilverMaster database gets corrupted before you get the CSR back from the certificate issuer. If you've saved the private key, you'll still be able to install the certificate.

1. Start the SMC.
2. Select the **Security** icon from the toolbar.
3. Select **Certificates**.
4. Select the **RSA** tab.
5. Choose **Export Private Key**. A message box displays advising you to use HTTPS.

   The following panel displays:



The RSA Private Key Export Wizard will export your private key to the path shown (in PKCS#8 format).

Edit the path if desired and be sure to record the location of this file. Press Finish when ready.

File name: C:\Program Files\Novell\exteNdAppServer\Certificate\rsaPKey_exp

Password:

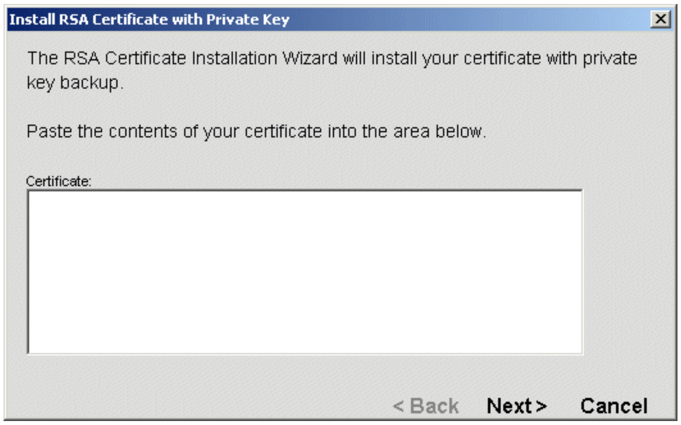
6. Supply the path and file name where you want to store the RSA private key.
7. Supply a password for this file. This does not have to be the administrator's password—it just applies to the file containing the private key information.
8. Click **Finish**.

➢ To install a certificate (with or without a private key):

1. Start the SMC.
2. Select the **Security** icon from the toolbar.
3. Select **Certificates**.
4. Select the **RSA** tab.
5. Choose **Install Certificate** or **Install with Private Key**.

   If you chose Install with Private Key, a message box displays and you are advised to use HTTPS for this procedure.

The following panel displays:



Install RSA Certificate with Private Key

The RSA Certificate Installation Wizard will install your certificate with private key backup.

Paste the contents of your certificate into the area below.

Certificate:

6 Paste the signed certificate into the text area and click **Finish**.

If you chose **Install with Private Key**, you are prompted for the file containing the private key and the password associated with the file:

6a Browse to the files location.

6b Enter the password and click **Finish**.

The SMC displays a message that the update was successful.

7 Click **OK**.

8 Choose **Restart** to have the changes take effect.

➢ To back up an RSA certificate:

1 Start the SMC.

2 Select the **Security** icon from the toolbar.

3 Select **Certificates**.

4 Select the **RSA** tab.

5 Choose **Export Certificate**. You are advised to run in HTTPS mode.

6 Specify the name and location of the backup file.

7 Specify a password to protect the file.

8 Click **Finish**.

➢ To import (install) an RSA certificate:

1 Start the SMC.

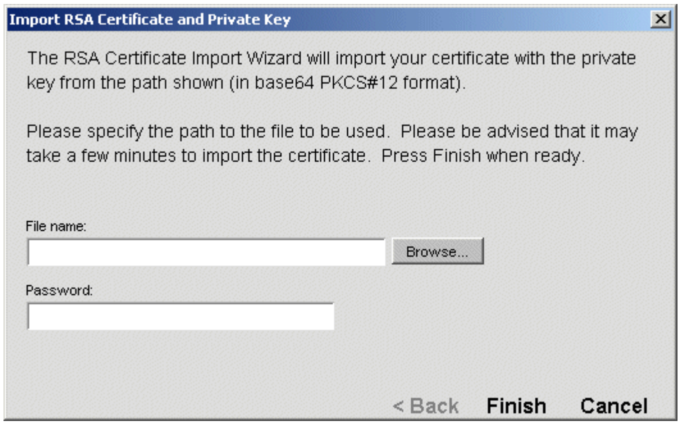
2 Select the **Security** icon from the toolbar.

3 Select **Certificates**.

4 Select the **RSA** tab.

5 Choose **Import Certificate**. You are advised to run in HTTPS mode.

The following panel displays:



Import RSA Certificate and Private Key

The RSA Certificate Import Wizard will import your certificate with the private key from the path shown (in base64 PKCS#12 format).

Please specify the path to the file to be used. Please be advised that it may take a few minutes to import the certificate. Press Finish when ready.

6 Specify the name and location of the certificate file.

7 Specify the password used to protect the file. (This is the same password used to export the certificate.)

8 Click **Finish**.

Generating and installing DSA server certificates

You can use the SMC to generate and install a DSA server certificate.

➢ To generate and install a DSA server certificate:

1 Start the SMC.

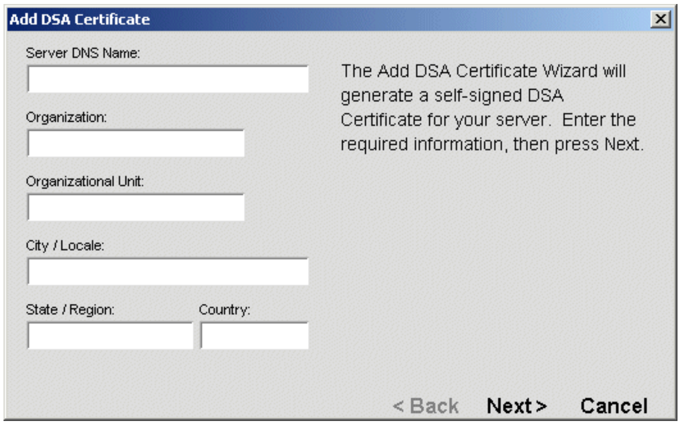
2 Select the **Security** icon from the toolbar.

3 Select **Certificates**.

4 Select the **DSA** tab.

5 Choose **Add Certificate**.

The following panel displays:



Add DSA Certificate

The Add DSA Certificate Wizard will generate a self-signed DSA Certificate for your server. Enter the required information, then press Next.

6 Complete the items on this panel as described below:

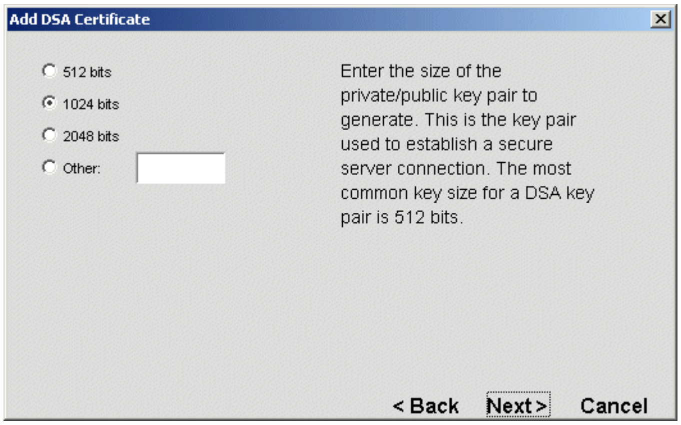
Field	What to enter
Server DNS Name	The TCP/IP hostname, which may be different from your machine name. (You can issue **ping localhost** from the command line to determine the TCP/IP name of the local host.)
Organization	Your company name in full unabbreviated legal form.

Field	What to enter
Organizational Unit	(Optional) Your department within the company.
City/Locale	(Optional) The city or locale where your company does business.
State/Region	The full name of the state or region where your company does business. (Do not abbreviate.)
Country	The country where your company does business. You must use the ISO two-letter country code. For example, the ISO code for the United States is US.

7 Click **Next**.

8 The following panel allows you to specify the size of the key pair to generate.

The 1024 bits option usually provides an acceptable level of security. Selecting a higher level decreases the speed of the initial connection. The 512 bits option provides a low level of security.



Add DSA Certificate

○ 512 bits
● 1024 bits
○ 2048 bits
○ Other:

Enter the size of the private/public key pair to generate. This is the key pair used to establish a secure server connection. The most common key size for a DSA key pair is 512 bits.

9 If prompted, specify the size of the key pair to generate.

10 Click **Next**.

You are warned that any existing certificate will be overwritten.

11 To continue adding the certificate, click **Finish**.

Creating and installing server certificates for the dispatcher

To create and install a server certificate for the application server's dispatcher (used in clustering), you must use these command-line utilities:

Utility	Description
AgDigitalIDStep1	Used to prepare either of the following: ◆ A CSR for an RSA certificate. The CSR is an encrypted file that contains information that identifies the organization running the application server. The CSR is submitted to a certificate issuer (such as VeriSign) that will use the information to create a certificate signed by the issuer. ◆ A self-signed DSA certificate that identifies the organization running the application server.
AgDigitalIDStep2	Prompts for the certificate information returned by the issuer as well as the private key used to protect the data, then installs the certificate (without private key information) into the application server.

The following sections include instructions for:

- Using AgDigitalIDStep1
- Using AgDigitalIDStep2

Using AgDigitalIDStep1

To enable HTTPS/SSL or TLS communications between application servers and clients in a clustered environment, you install an RSA or DSA certificate on the application server’s Dispatcher.

➢ To generate an RSA or DSA server certificate:

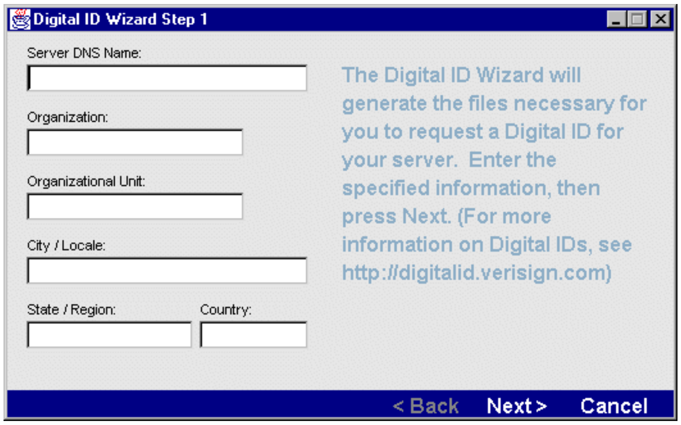
1 Change the working directory to the server’s **\bin** directory.

2 At the command line, specify one of the following commands:

Certificate type	Command
RSA certificate	`AgDigitalIDStep1`
DSA certificate	`AgDigitalIDStep1 dsa`

TIP: The titles and some of the help text may differ slightly if you are generating a DSA certificate.

The following panel displays:



The Digital ID Wizard will generate the files necessary for you to request a Digital ID for your server. Enter the specified information, then press Next. (For more information on Digital IDs, see http://digitalid.verisign.com)

TIP: In UNIX, this utility runs using a GUI and cannot be run in a character terminal window. If you log in remotely to the UNIX machine, make sure you set your DISPLAY environment variable appropriately.

3 Complete the items on this panel as follows:

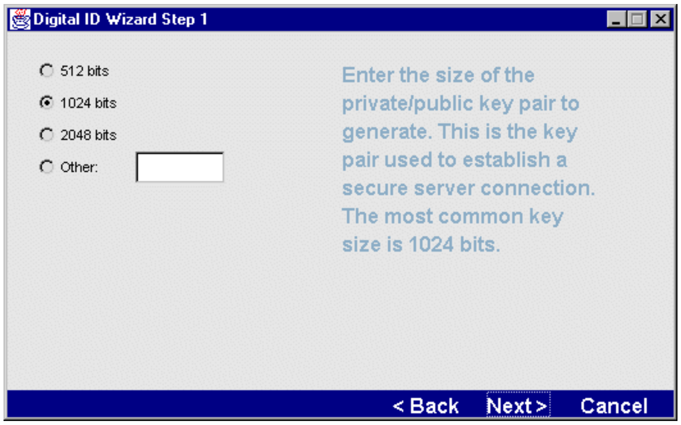
Field	What to specify
Server DNS Name	The TCP/IP hostname, which may be different from your machine name. (You can issue **ping localhost** from the command line to determine the TCP/IP name of the local host.)
Organization	Your company name in full unabbreviated legal form.
Organizational Unit	(Optional) Your department within the company.
City/Locale	(Optional) The city or locale where your company does business.
State/Region	The full name of the state or region where your company does business. Do not abbreviate.

Field	What to specify
Country	The country where your company does business. You must use the ISO two-letter country code. For example, the ISO code for the United States is **US**.

4 Click **Next**.

The following panel displays for you to specify the size of the key pair to generate.

The 1024 bits option usually provides an acceptable level of security. Selecting a higher level decreases the speed of the initial connection. The 512 bits option provides a low level of security.



○ 512 bits
● 1024 bits
○ 2048 bits
○ Other:

Enter the size of the private/public key pair to generate. This is the key pair used to establish a secure server connection. The most common key size is 1024 bits.

5 If prompted, specify the size of the key pair to generate.

6 Click **Next**.

6a If you are generating an RSA certificate, enter a password then confirm it. This password will be used to encrypt your private key.

TIP: Be sure to record this password; you will need it to install the certificate.

6b Click **Next**.

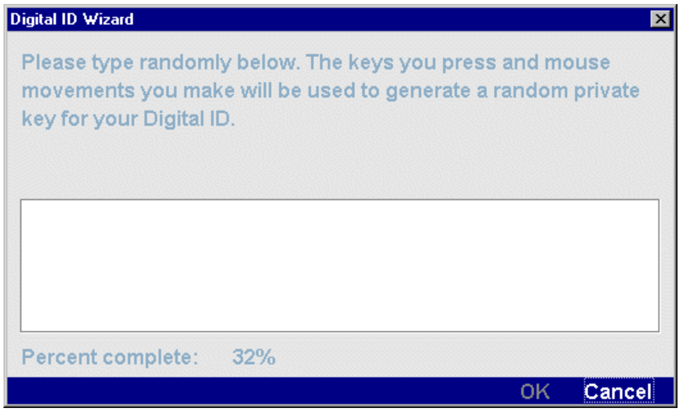
7 **For RSA certificates**, the panel shows the paths for the CSR and the password-protected private key.

For DSA certificates, the panel shows the paths for the certificate and the private key files. You may edit these paths if you choose. You will use this information later when installing the certificate.

IMPORTANT: The file that contains the private key must be kept physically secure. Otherwise, anyone who can obtain the server’s certificate can masquerade as the server.

8 Click **Next**.

If the wizard hasn't been able to collect enough randomness information to generate cryptographically good keys from the key presses and mouse movements you have made, the following panel displays:



Please type randomly below. The keys you press and mouse movements you make will be used to generate a random private key for your Digital ID.

Percent complete: 32%

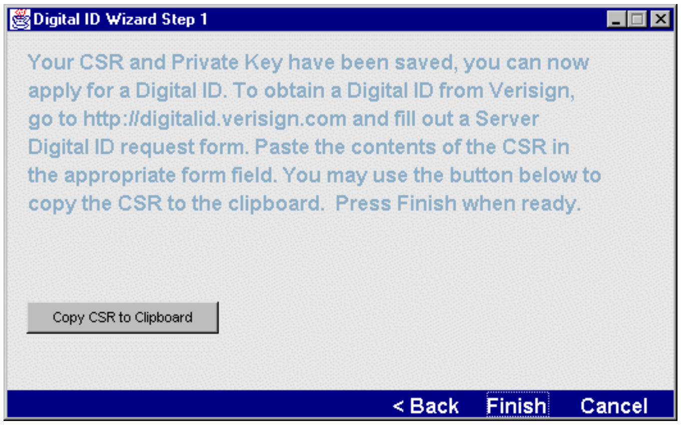
9 If prompted, type random characters in the edit box and move the mouse around to create an encrypted private key. When the wizard has enough randomness information, the **OK** button is enabled.

For RSA certificates, the wizard generates the certificate signing request and private key.

For DSA certificates, the wizard generates the certificate.

10 Click **OK**.

The following panel displays:



Your CSR and Private Key have been saved, you can now apply for a Digital ID. To obtain a Digital ID from Verisign, go to http://digitalid.verisign.com and fill out a Server Digital ID request form. Paste the contents of the CSR in the appropriate form field. You may use the button below to copy the CSR to the clipboard. Press Finish when ready.

11 **For RSA certificates**, click **Copy CSR to Clipboard** to copy the contents of the CSR to the clipboard for use in the next step.

For DSA certificates, click **Copy Digital ID to Clipboard** for use when installing the certificate.

12 **For RSA certificates**, follow the directions to request a certificate for your application server (for example, by using the VeriSign Web site http://digitalid.verisign.com). Once your request is approved, the certificate authority sends the new certificate back to you through e-mail.

13 Click **Finish**.

You use AgDigitalIDStep2 to install the certificate. See "Using AgDigitalIDStep2" next.

Using AgDigitalIDStep2

Once you receive your RSA certificate from the CA or have generated a DSA certificate using AgDigitalIDStep1, you can install the certificate using AgDigitalIDStep2.

➢ To install an RSA or DSA certificate:

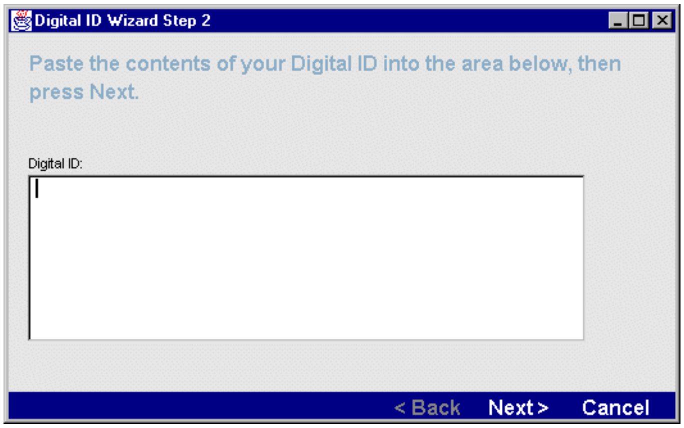
1 Start the Dispatcher using the -c (upload certificate) option.

2 Change the working directory to your server's **bin** directory.

3 At the command line, specify the following command:

```
AgDigitalIDStep2
```

The following panel displays:



Paste the contents of your Digital ID into the area below, then press Next.

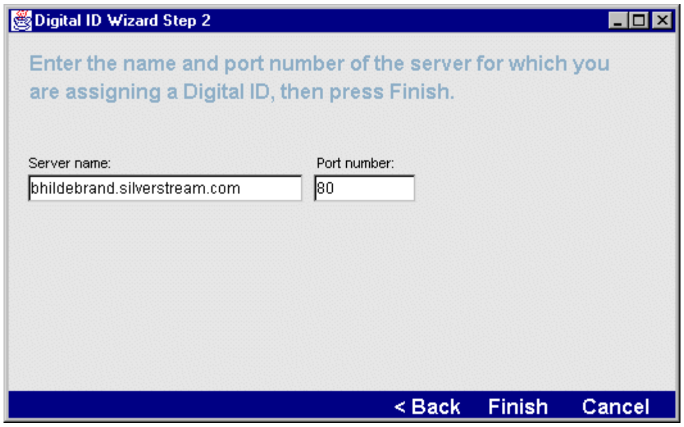
4 Paste the certificate into the panel.

5 Click **Next**.

6 This panel asks for the path for the private key (you may need to edit the name of the private key, since the names are different for RSA and DSA certificates) and asks you to confirm your password (there is no password for a DSA private key).

7 Click **Next**.

The following panel displays:



Enter the name and port number of the server for which you are assigning a Digital ID, then press Finish.

Server name: bhildebrand.silverstream.com

Port number: 80

8 Enter the Dispatcher's name (in the server name text field) and the Dispatcher's HTTP port number.

If you have configured separate ports for different types of operations, specify your **administration** port. By default, the application server listens to port 80.

9 Click **Finish**.

A confirmation message displays.

10 To activate the certificate, click the **Restart** (server) button on the SMC.

After the server restart, the server is configured to listen:

- For **HTTP** requests on the HTTP port.
- For **HTTPS** requests at the RSA or DSA port (depending on the kind of certificate you just installed).

For more information, see "Enabling RSA/DSA ports" on page 149 and "Specifying general server properties" on page 80.

When you are ready for production, use the SMC to enable authorization. See "Enabling authentication" on page 159.

Viewing server certificates

➢ To view certificates that have been installed on the server:

1 Start the SMC.

2 Select the **Security** icon from the toolbar.

3 Select **Certificates**.

4 Select the **Certificate List** tab.

5 Choose a certificate from the dropdown.

Enabling RSA/DSA ports

By default, the application server specifies port 443 for RSA and DSA communications. You enable and modify ports for runtime and administrative access for each of the following three security protocols: HTTP, HTTPS-RSA, and HTTPS-DSA. The server does not require you to configure unique port values for the different types of access; ports having the same value will share the same socket and will allow multiple operations.

For more information, see "Setting up separate ports" on page 79.

➢ To enable and change RSA/DSA ports:

1 Start the SMC.

2 Select the **Security** icon from the toolbar.

3 Select **Certificates** and choose either the **DSA** tab or the **RSA** tab.

4 In the Port Settings section of the tab, under Port Settings, select any of the check boxes to enable **Runtime** or **Admin** ports for DSA or RSA:

   - Enable RSA only after you have installed an RSA server certificate from a provider such as VeriSign.
   - Enable DSA only if you have installed a DSA certificate on the server.

   TIP: After a server certificate has been installed, the ports are automatically enabled when the server is restarted.

   For more information, see "Creating and installing server certificates for the dispatcher" on page 144.

5 Change the RSA and DSA port numbers of the **Runtime** and/or **Admin** ports if necessary.

   With a UNIX server, specify port numbers above 1024 if the server is not being run with root access (ports below 1024 are reserved for root access).

   For information about cipher suites, see "Restricting SSL cipher suites" on page 151.

6 Select **Update**.

7. To activate the changes, click **Restart** (server).

   Once ports are enabled, Java clients will verify server certificates when establishing SSL connections to the application server.

   📖 For more information, see "Verifying SSL server certificates for Java clients" on page 158.

Turning off HTTP communications

You can turn off HTTP communications and allow client communications using only HTTPS or RMI.

📖 For more information, see "Specifying ORB settings" on page 84.

If you accidentally disable ports that prevent you from running the SMC (for example, if you disable all the administration ports), you will need to edit the httpd.props file to re-enable the administration port. You can enable and disable runtime ports using the SMC.

➢ To disable HTTP communications:

1. Start the SMC.
2. Select the **Configuration** icon from the toolbar.
3. Select **General**.
4. In the HTTP Ports section, deselect the **Enable HTTP Runtime Port** (or the HTTP Admin ports) check box to disable.

   The server will run even if you disable all runtime ports. You must be careful how you disable runtime ports. When you disable HTTP runtime ports, the server checks to see if you have DSA or RSA runtime ports enabled. It warns if you proceed to disable the HTTP runtime port.
5. Click **Update**.
6. To activate the change, click the **Restart** button. For more information, see "Restarting the application server" on page 74.

Now the server will not listen on the HTTP server port.

Allowing SSL 3.0 and TLS 1.0

The application server supports SSL (Secure Sockets Layer) 3.0 and TLS (Transport Layer Security) 1.0. TLS 1.0 protocol is based on the SSL 3.0 protocol, and is sometimes referred to as SSL 3.1.

By default, SSL 3.0 and TLS 1.0 are both allowed

➢ To allow or disallow SSL 3.0:

1. Start the SMC.
2. Select the **Security** icon from the toolbar.
3. Select **General**.
4. Check the **SSL 3.0** checkbox to enable SSL 3.0

   To run in FIPS-mode, TLS 1.0 must be enabled and SSL 3.0 must be disabled.
5. Click **Update**.
6. Restart your server.

➢ To allow or disallow TLS 1.0:

1. Start the SMC.
2. Select the **Security** icon from the toolbar.
3. Select **General**.

4 Check the **TLS 1.0** checkbox to enable TLS1.0

To run in FIPS-mode, TLS 1.0 must be enabled and SSL 3.0 must be disabled.

5 Click **Update**.

6 Restart your server.

Restricting SSL cipher suites

When an SSL connection is initialized, the browser client and the server determine a common cipher value to be used for key exchange and encryption. Various cipher values offer different types of encryption algorithms and levels of security. The application server has a full set of cipher suites that can service a range of clients by providing low, medium, and high-level encryption.

You can restrict the levels of encryption (cipher values) used by the server when communicating in HTTPS through the RSA and DSA runtime ports. This allows you to have a server capable of high-level encryption while preventing connections from lower-level security clients.

During startup, the application server reads the list of allowed cipher suites. See "Restricting SSL cipher suites" on page 151 for the list of cipher suites that are enabled by default. You can use the SMC to change the enabled/disabled cipher suites. Only the selected cipher suites will be used to initialize the appropriate SSL socket(s).

➢ **To specify which cipher suites are allowed:**

1 Start the SMC.

2 Select the **Security** icon from the toolbar.

3 Select **Certificates**.

4 Choose the **RSA** or **DSA** tab (depending on the type of certificate that is installed).

5 Select the **Enable Runtime Port** check box to enable secure DSA (Java-client) and/or RSA (HTML and Java-client) communications.

6 Specify which cipher suites are allowed by clicking **Cipher Suites** and selecting and deselecting the cipher suites in the resulting panels, as described in the table below. When you select a cipher suite, its description displays in the panel.

7 Click **OK** to exit the Cipher Suites dialog and accept your changes.

8 Click **Update**.

9 To activate the changes, click **Restart** (server).

Table of cipher suites The following table lists the cipher suites (and levels of security) supported by the application server when using secure communications (HTTPS).

Certificate type	AES	Encryption strength (bits)	Cipher suites
DSA	N	40	SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA*
		56	TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA*
			TLS_DHE_DSS_EXPORT1024_WITH_RC4_56_SHA*
			SSL_DHE_DSS_WITH_DES_CBC_SHA*
		128	TLS_DHE_DSS_WITH_RC4_128_SHA*
		168	SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA*
	Y	128	TLS_DHE_DSS_WITH_AES_128_CBC_SHA*
		256	TLS_DHE_DSS_WITH_AES_256_CBC_SHA *
RSA	N	0	SSL_RSA_WITH_NULL_MD5
			SSL_RSA_WITH_NULL_SHA
		40	SSL_RSA_EXPORT_WITH_RC4_40_MD5 *
			SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5*
			SSL_RSA_EXPORT_WITH_DES_40_CBC_SHA*
			SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA*
		56	TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA*
			TLS_RSA_EXPORT1024_WITH_RC4_56_SHA*
			SSL_RSA_WITH_DES_CBC_SHA*
			SSL_DHE_RSA_WITH_DES_CBC_SHA*
		128	SSL_RSA_WITH_RC4_128_MD5*
			SSL_RSA_WITH_RC4_128_SHA*
		168	SSL_RSA_WITH_3DES_EDE_CBC_SHA*
			SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA*
	Y	128	TLS_RSA_WITH_AES_128_CBC_SHA*
			TLS_DHE_RSA_WITH_AES_128_CBC_SHA*
		256	TLS_RSA_WITH_AES_256_CBC_SHA *
			TLS_DHE_RSA_WITH_AES_256_CBC_SHA *
None	N	40	SSL_DH_anon_EXPORT_WITH_DES_40_CBC_SHA
			SSL_DH_anon_EXPORT_WITH_RC4_40_MD5
		128	SSL_DH_anon_WITH_RC4_128_MD5
		168	SSL_DH_anon_WITH_3DES_EDE_CBC_SHA
	Y	128	TLS_DH_anon_WITH_AES_128_CBC_SHA
		256	TLS_DH_anon_WITH_AES_256_CBC_SHA

* A cipher suite that is enabled, by default, at server startup.

Managing Certificate Authorities

The application server maintains a list of Certificate Authorities (CAs) for verifying client certificates. This represents the list of guarantors the server will trust. There are three common CAs installed when the server is initially configured. These CAs are from VeriSign, Inc. and represent different levels of trust: class 1 represents a certificate that has minimal trust; class 3 represents the highest level of trust.

When a user with a client certificate tries to access the server, the server first checks the list of CAs to verify that the certificate has been approved by a known party, then checks for a valid timestamp to verify that the certificate has not expired. After completing the verification, the server handles the connection request according to the Client Certificate Level parameter (which you can set using the SMC; see "Installing and managing client certificates" on page 153).

The application server can also server extract the CA from an unrecognized client certificate. For more information, see "Installing and managing client certificates" on page 153.

➢ **To install or delete a CA certificate:**

1. Start the SMC.
2. Select the **Security** icon from the toolbar.
3. Select **Certificates**.
4. Select the **Authorities** tab.

   This tab contains information about each CA and allows you to add new CAs to allow a broader range of clients to be trusted or to delete CAs to tighten the level of security on the server.
5. To add a CA, click **Add Certificate Authority** and select the file. To delete a CA, highlight it and click **Delete Certificate Authority**.

Installing and managing client certificates

In the HTTPS environment, a client certificate establishes the identity of a user when communicating with a server. Client certificates can be obtained from various sources, but to be useful the certificate must be signed by a guarantor (Certificate Authority or CA) trusted by the server.

The application server supports client certificates from standard Internet browsers (including Netscape and Internet Explorer) that use RSA encryption and supports X.509 certificates (which is a particular implementation of the Certificate interface used by many certificate issuers).

Client certificates and EJBs Client certificates that use RSA encryption are also used to establish the identity of a user in the SSL environment (used by EJBs). As with the HTTPS environment, the server supports client certificates from standard Internet browsers. If you need to add a new CA to the server, see "To install or delete a CA certificate:" on page 153.

Enabling and installing client certificates

You can use the SMC to determine how the server will handle connection attempts from users with valid client certificates. Seven parameter options are available, each representing a different level of restriction. Two of the options will automatically install new certificates that are verified by the server and add them to the database as new users in the Certificate Security realm. You can also install the certificates manually (see "Manually installing client certificates" on page 155).

Each set of HTTPS ports (HTTPS-RSA HTTPS-DSA) has a single set of cipher suites associated with it. The cipher suite you select applies to all ports (runtime and administration) of that type.

NOTE: The SMC will not allow HTTPS ports to be enabled without a valid certificate. If you try to enable an HTTPS port (using the SMC or by editing the props file) without first installing a certificate, the server error will display on startup. See “The httpd.props File” on page 237.

➢ **To enable client certificates on the server:**

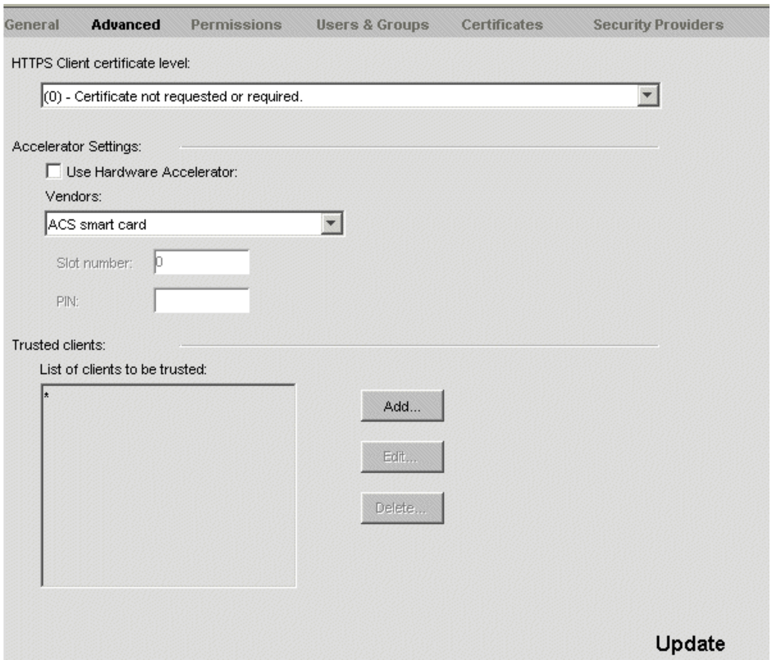
1 Start the SMC.

2 Select the **Security** icon from the toolbar.

3 Select **Advanced**.

Each set of HTTPS ports (HTTPS-RSA HTTPS-DSA) has a single set of cipher suites associated with it. Although the cipher suite you select applies to all ports (runtime and administration) within a protocol type, you can set each port to a different value.

IMPORTANT: You cannot enable an HTTPS port without first installing an HTTPS-RSA or HTTPS-DSA server certificate. If you try to enable an HTTPS port (using the SMC or by editing the props file) without first installing a certificate, the server error will display on startup.



HTTPS Client certificate level: (0) - Certificate not requested or required.

Accelerator Settings:

□ Use Hardware Accelerator:

Vendors: ACS smart card

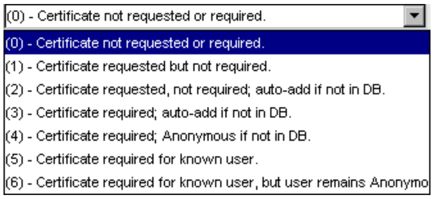
Slot number: 0

PIN:

Trusted clients:

List of clients to be trusted: *

4 Select an option from the dropdown list labeled **HTTPS Client certificate level**, then click **Update**:



(0) - Certificate not requested or required.

(0) - Certificate not requested or required.
(1) - Certificate requested but not required.
(2) - Certificate requested, not required; auto-add if not in DB.
(3) - Certificate required; auto-add if not in DB.
(4) - Certificate required; Anonymous if not in DB.
(5) - Certificate required for known user.
(6) - Certificate required for known user, but user remains Anonymo

You need to restart the server **only** if you changed from level 0 to another level.

Your selection determines how the server will handle all valid client certificates. The certificate verification table (below) describes each option. The options are numbered 0 (no verification) to 6 (most restrictive level).

Certificate verification table

Certificate verification level	Description
Certificate not requested or required (0)	Turns off the use of certificates for establishing a client's identity. A client that connects via HTTPS will still benefit from secure communications and can check the identity of the server, but will not be asked to present its own certificate.
Certificate requested but not required (1)	The server will request a certificate from the client. If the client does not have a certificate or has a certificate that has not been validated by the server, a connection is allowed but the user remains anonymous. If a certificate is presented for an established user, the client takes on the identity of the certificate user.
Certificate requested, not required; auto-add if not in database (2)	The server will request a certificate from the client. If the client does not have a certificate, a connection is allowed but the user will remain anonymous. If a legitimate certificate is presented but does not yet correspond to a client validated by the server, the server will automatically add the certificate, and the client takes on the identity of the newly created Certificate Security user. If a certificate is presented for an established user, the client takes on the identity of the previously established Certificate Security user.
Certificate required; auto-add if not in database (3)	The server will request a certificate from the client. If the client does not have a certificate, the connection is denied. If a legitimate certificate is presented but does not yet correspond to a client validated by the server, the server will automatically add the certificate, and the client takes on the identity of the newly created Certificate Security user. If a certificate is presented for an established user, the client takes on the identity of the previously established Certificate Security user.
Certificate required; Anonymous if not in database (4)	The server will request a certificate from the client. If the client does not have a certificate, the connection is denied. If a legitimate certificate is presented but does not yet correspond to a client validated by the server, the client is allowed to connect but remains Anonymous. If a certificate is presented for an established user, the client takes on the identity of the previously established Certificate Security user.
Certificate required for known user (5)	The server will request a certificate from the client. If the client does not have a certificate or has a certificate that does not correspond to a client previously validated by the server, the connection is denied. If a certificate is presented for an established user, the client takes on the identity of the previously established Certificate Security user.
Certificate required for known user, but remain Anonymous (6)	The server will request a certificate from the client. If the client does not have a certificate or has a certificate that does not correspond to a client previously validated by the server, the connection is denied. If a certificate is presented for an established user, the client is allowed to connect but still remains Anonymous.

Manually installing client certificates

The previous section describes parameter options that automatically add new validated certificates to the database on the server. You may prefer to install the certificates separately. This section describes how to manually add client certificates to the database. It consists of two procedures, one for the client machine and the other for the server.

➢ To install a client certificate—client machine:

NOTE: For this procedure to work, the certificate validation level on the server must be set to 1, 2, or 4 (see “Certificate verification table” on page 155).

1 Make sure you have a valid client certificate installed on your browser.

2 Open your browser and go to the following URL:

```
https://server/SilverStream/Meta/Certificates?action=data
```

where *server* is the name of your server.

The server will extract the user information from the certificate and send it back to the client.

3 Save the file that the server presents to an appropriate area.

➢ To install a client certificate—server machine:

1 Start the SMC.

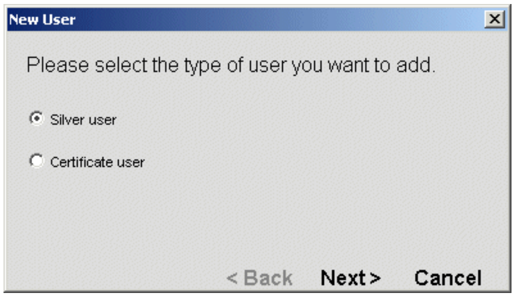
2 Select the **Security** icon from the toolbar.

3 Select **Users & Groups**.

4 Select **Certificate Security**.

5 Choose the **New User** icon at the bottom of the right pane:

The following panel displays:



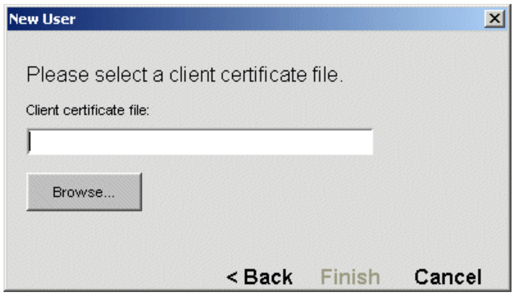
New User

Please select the type of user you want to add.

● Silver user

○ Certificate user

6 Choose **Certificate user**:



New User

Please select a client certificate file.

Client certificate file:

7 Enter the file name obtained by the client from the previous procedure (“To install a client certificate—client machine:” on page 156) and click **Finish**.

This adds the certificate as a new user to the database on the server.

Extracting a CA from a client certificate

The application server can extract the CA from a client certificate if the CA is not installed on the server. This section consists of two procedures, one for the client machine and the other for the server.

➢ To extract the CA from a certificate—client machine:

NOTE: For this procedure to work, the certificate validation level on the server must be set to 1 or 2 (see “Certificate verification table” on page 155).

1. Make sure you have a valid client certificate installed on your browser.
2. Open your browser and go to the following URL:

   ```
   https://server/SilverStream/Meta/Certificates?action=dataCA
   ```

   where *server* is the name of your server.

   The server will extract the CA from the certificate and send it back to the client.
3. Save the file that the server presents to an appropriate area.

➢ To extract a CA from a client certificate—server:

1. Start the SMC.
2. Select the **Security** icon from the toolbar.
3. Select the **Certificates** panel.
4. Select the **Authorities** tab.
5. Click **Add Certificate Authority**.
6. Browse for the location of the extracted CA, then click **Open**.

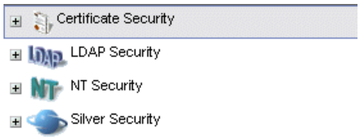
   The program installs the new CA to the database.
7. To activate the new CA, restart the server by clicking the **Restart** (server) icon.

Accessing certificate users

Certificate users are added to a security realm called Certificate Security. This realm is included in the lists of users and groups supported by the application server.

➢ To access certificate users:

1. Start the SMC.
2. Select the **Security** icon from the toolbar.
3. Select **Users & Groups**.
4. Navigate to a user from the **Certificate Security** list:



5. Use this panel to view current users or add users to groups.

   📖 For more information about users and groups, see “Setting Up Users and Groups” on page 89.

You can use certificate users in security expressions exactly as you use other Silver Security users. For more information about security expressions, see “Authorization and access control” on page 165.

Updating client certificates

An X.509 certificate includes a start date and an end date. A client is not allowed to connect to the server with a certificate that has expired. You can update the certificate representing an established Certificate Security user by obtaining a new certificate and assigning it to a user already known to the application server. This ensures that any security expressions involving the existing user will continue to work properly.

Normally, an existing certificate user would present the updated certificate to the server using the URL described in "Manually installing client certificates" on page 155.

NOTE: If you are running the server with one of the autoinstall parameters described previously (level 2 or 3) and an existing client certificate holder attempts to access the server with a new (updated) certificate that the server validates, that client will be installed as a new user. This client may be restricted from resources available to the previous user.

➢ **To update an existing certificate user:**

1. Select the **Security** icon from the toolbar.
2. Select the **Users & Groups** panel.
3. From the Certificate Security domain, navigate to the user you want to update.
4. Click the **Properties** button.

   A form with three tabs displays. The General tab shows general information concerning the user. The Additional Attributes tab displays information about the certificate. The Update tab allows you to update the certificate for the selected user.
5. Select the **Update** tab.
6. Enter the file name for the updated certificate, then click **Update**.

   This replaces the old certificate with the new version without changing the identity of the user on the server.

Verifying SSL server certificates for Java clients

Java clients verify server certificates against a list of trusted CA certificates (stored in the agrootca.jar file) when establishing an SSL connection to the application server. If a certificate cannot be verified, the SSL connection cannot be established.

To use SSL communications, you must have one of the following server certificates installed on your application server:

Certificate	Details
An **RSA certificate purchased** from an external company that offers certificate services (such as VeriSign)	You use the SMC or the command-line tool (AgDigitalIDStep1) to generate an RSA Certificate Signing Request (CSR). The encrypted CSR file is submitted to an external CA, which identifies the organization running the application server and creates a signed certificate. If you use an external CA and RSA encryption, see "Using the agrootca.jar file to verify RSA certificates" on page 159.
A **self-generated RSA certificate** (you act as your own CA using your internal organization's tool, such as Netscape Certificate Server)	If you generate your own certificate, you need to distribute the agrootca.jar file as described in "Using the agrootca.jar file to verify RSA certificates" on page 159.
A **self-signed DSA certificate** created using the SMC	Organizations using a self-signing DSA server certificate should use the command-line option, as described in "Command-line option for self-signing DSA server certificates" on page 159.

Using the agrootca.jar file to verify RSA certificates

Clients verify the application server's certificate against a list of trusted CA certificates that are stored in the **agrootca.jar** file. At startup, the client reads in the list of CAs and then checks all server certificates against the contents of the agrootca.jar file to verify that certificates are signed.

You should include all trusted root CAs in the agrootca.jar file and remove any untrusted CAs. Clients will trust only those server certificates that were signed by the root CA certificates in the JAR file.

If you generate your own RSA certificates using a tool such as Netscape Certificate Server, you should put the CA certificate of the Certificate Server into the agrootca.jar file and then distribute the JAR file to each client machine running SilverJ2EEClient. The agrootca.jar file should be stored in SilverJ2EEClient's **Common\lib** directory.

NOTE: The agrootca.jar file is used by the application server only when user code tries to connect to another server using SSL (such as in the case of a servlet).

Simplified secure port configuration You can use the application server's RSA port to provide secure communications between the server and Java clients, HTML clients, and EJBs.

Command-line option for self-signing DSA server certificates

If your organization uses self-signed DSA certificates to provide data encryption between the application server and clients, you should use the **+Dsssw.ssl.nocacheck** command-line option when you run SilverJ2EEClient. This option prevents the client from attempting to verify the certificate authority. For example:

```
SilverJ2EEClient +Dsssw.ssl.nocacheck server database warfile
```

Enabling authentication

You can require user authentication and activate related settings. You set authentication settings when you are ready to deploy applications that require authentication for both HTTP user/password protection and HTTPS-RSA client certificates.

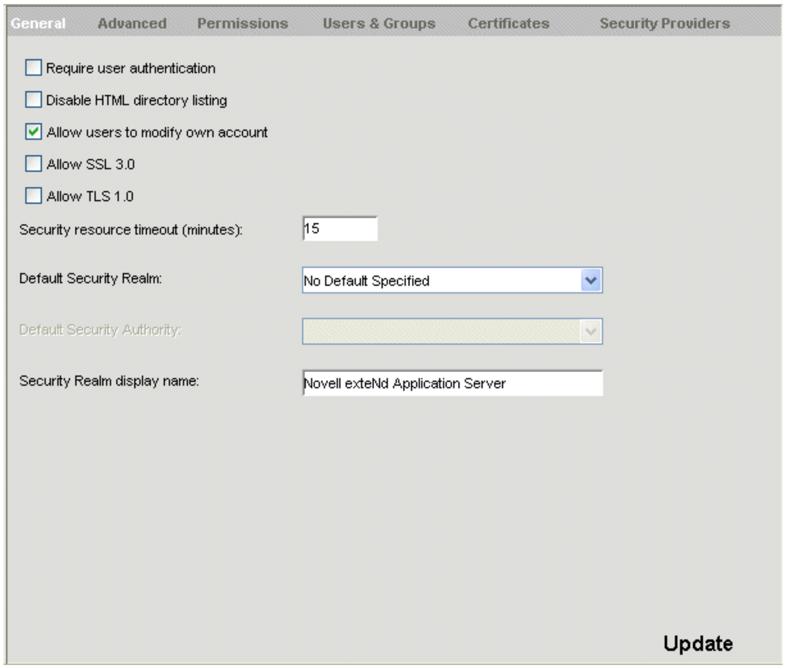
📖 For more information, see "Setting up separate ports" on page 79.

➢ **To enable user authentication:**

1 Start the SMC.

2 Select the **Security** icon from the toolbar.

3 Select **General**:



☐ Require user authentication

☐ Disable HTML directory listing

☑ Allow users to modify own account

☐ Allow SSL 3.0

☐ Allow TLS 1.0

Security resource timeout (minutes): 15

Default Security Realm: No Default Specified

Default Security Authority:

Security Realm display name: Novell exteNd Application Server

4 Activate the security options as follows:

Item	Description
Require user authentication	Requires users when first accessing the server to authenticate themselves, either through a certificate or user name/password. Once authenticated, users do not have to be authenticated again during the session. Setting this option disallows Anonymous users; if you don't set this option, a user can access the server without logging in or sending a certificate, in which case the user is known as Anonymous. If you are using this option with HTTPS, activate **Client certificate level in HTTPS**. You can also require login on a per-object basis. See "Changing access" on page 168.
Disable HTML directory listing	Disables the listing of server directory contents in a browser when the browser points to the directory's URL. Prevents users from seeing directory contents. If this option is turned on, the server returns a FORBIDDEN error.
Allow users to modify own account	By default, users can change their own user properties. You can turn off this privilege by deselecting this check box. If this privilege is turned off, only administrators (users with Read Server Configuration and Modify Server Configuration permissions) can change user properties. For more information, see "Editing user properties" on page 91.
Security resource timeout	Specifies how frequently the application server will upload current lists of users and groups from the NT, LDAP, and/or NIS+ servers, which would include any updates to these lists. For more information, see "Resetting the security resource timeout" on page 126.

5 For changes to take effect, click **Update**.

Using Cryptographic Hardware Integration

Cryptographic Hardware Integration (CHI) enables significant application server SSL encryption/decryption performance enhancements on machines with the application server and a supported hardware accelerator card.

📖 For information about the supported cards, see the application server *Release Notes*.

To use CHI, you need to:

1. Install a supported hardware accelerator card on the machine where the application server is installed

   📖 For information about installing and setting up the card, see your hardware card's documentation.

2. Install CHI on the machine (see "To install CHI:" below)
3. Configure the server using the SMC (see "To configure the application server to use the hardware accelerator card:" below)

➢ To install CHI:

1. Invoke **chiVersionInstall.exe** (the CHI installer).
2. Follow the prompts to install CHI into your application server's installation directory.

You can later uninstall CHI separately if you want.

➢ To configure the application server to use the hardware accelerator card:

1. Start the **SMC**.
2. Select the **Security** icon from the toolbar.
3. Select **Advanced**.
4. Select **Use Hardware Accelerator**.
5. Specify your accelerator card.
6. Specify the slot number and PIN, which you can get or set up from the utility software that came with the card.
7. Click **Update**.
8. Restart the server.

Managing trusted clients

You can use the SMC to set up a list of clients that are trusted by an application server when receiving an asserted identity in EJB calls. For standalone servers, this is a server-level property. For servers running in a cluster, this is a cluster-level property—any trusted clients set up on a single server are propagated to all servers in the cluster.

➢ To add a client to the list of trusted clients:

1. Start the **SMC**.
2. Select the **Security** icon from the toolbar.
3. Select **Advanced**.
4. In the Trusted Clients section of the Advanced tab, choose **Add**.

   You are prompted for the client name.

5. Enter the client's host name.

   The entry can contain the asterisk wild card character(*)—for example:

```
*.mydomain.com
```

OR

```
server*.mydomain.com
```

OR

```
*
```

If you use the asterisk, it should be the last character in a section of the URL. Any characters following the asterisk in a section are ignored. In the example shown above, if server* were shown as server*1, the 1 would not be included in the list.

Configuring FIPS-compliant mode

This section describes how to configure the application server to run in the Federal Information Processing Standards (FIPS) mode. The application server is FIPS-ready by default. You need to follow the procedures below to FIPS-enable the application server, the dispatcher, and client connections (these includes outbound connections from the application server, SMC, or SilverJ2EEClient).

NOTE: Silver Security is not FIPS-compliant. To be FIPS-compliant, you must use LDAP or another user registry mechanism for managing users and passwords.

➢ To configure the application server to run in FIPS-approved mode:

1 Use the SMC to disable SSL 3.0 and enable TLS 1.0.

📖 For more information, see "Allowing SSL 3.0 and TLS 1.0" on page 150

2 Use the SMC to disable all of the cipher suites except:

- SSL_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_WITH_DES_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA

📖 For more information, see "Restricting SSL cipher suites" on page 151

➢ To run SilverCmd, SilverJ2EEClient, and the SMC in FIPS-approved mode:

1 Disable SSL 3.0 and enable TLS1.0:

1a Open ciphersuitesclients.props file (located in the \Common\lib).

1b To disable SSL 3.0, make sure that the file includes the uncommented line:

```
NOSSL3
```

1c To enable TLS, make sure the NOTLS1 command is removed from the file or commented like this:

```
#NOTLS1
```

📖 For more information about the configuration file, see"About the cipher suite configuration file" on page 163.

➢ To configure the dispatcher to run in FIPS-approved mode:

- Modify Common/lib/ciphersuites.props

or

- Supply your own configuration file using the -ciphersuites command line option at startup.

📖 For more information about the configuration file, see"About the cipher suite configuration file" on page 163.

Other FIPS-compliance considerations:

- **Sun’s JSSE and JCE implementations** are not FIPS-approved. Using them in a deployed application will cause the server to be noncompliant.
- **If your application uses URL or URLConnection**, you will very likely remain in compliance. The URLStreamHandlerFactory uses Phaos SSLava which is based on a Phaos Crypto module. However, if you call URL.setURLStreamHandlerFactory() to another implementation, then you will become noncompliant.

About the cipher suite configuration file

The cipher suite configuration file is a Java properties file. The legal entries are:

- NOSSL3—When present, disallows SSL 3.0
- NOTLS1—When present, disallows TLS 1.0
- NONTLM— When present, disallows NTLM authentication in Composer
- *ciphersuitename=Priority#*—An optional integer

The cipher suite names are sorted descending according to priority#. If no priority# is specified, the cipher suite has the lowest priority.

The application server does not support all of the alias names of cipher suites. (See “Restricting SSL cipher suites” on page 151 for the list of supported cipher suites.) The cipher suite configuration files are located in the Common/lib folder or in a JAR file:

- For client side, the file is named ciphersuitesclient.props
- For server side, the file is named ciphersuites.props

10 Using Security

This chapter describes ways to provide and restrict access to deployment databases, application servers, and application server clusters. This chapter contains sections on:

Authorization and access control

Access control specifies what operations users are authorized to perform on objects and resources.

Authorization and access control

The application server's authorization and access control settings manage access to:

- The server
- Deployed applications
- Application data

The application server uses *security expressions* to determine what an identified user is allowed to access.

The application server supports the Java security Access Control List (ACL) API, allowing developers to programmatically control access to data and deployed applications. This Java security works with the security provided by the SMC and the application server's Server Administration API.

📖 For more information about the Server Administration API, see Chapter 13, "Using the Server Administration API".

You can use the SMC to control access to these resources:

- Server directories
- J2EE archives deployed to the server
- Media objects stored on the server

You specify access by setting the permissions described in "Permission types" on page 166.

Permission types

You use permissions to set up access control on various **administration operations** and **database objects**.

Administrative server permissions

The **administration resource** controls your ability to view and modify administrative settings (and permissions to access settings). Once you have secured the administration resource, unauthorized users will not be able to perform administrative operations.

NOTE: Only members of the Administrators group can access administration settings. To prevent unauthorized users from accessing all administrative information such as who is logged in, session information, permission settings for users and groups, number of connections, and so on, it is important to retain this restriction.

Each of the permission tabs listed in the following table represents an aspect of the administration resource that should remain restricted:

Permission	Description
Read Server Configuration	Limits who can view administration settings and connection parameters.
Modify Server Configuration	Limits who can administer the server or cluster by changing connection parameters such as who is logged in, permission settings for users and groups, number of connections, and so on.
Read Directory Listing	Limits who can browse the server directories. Does **not** affect whether a user can access an object.
Read Users and Groups	Limits who can see users and groups that have been defined on the server. Users must have this permission in order to set permissions; otherwise, they cannot see the users and groups.
Set Permissions	Limits who can change the permissions that control the administration resource. Anyone with Locksmith privilege will bypass the check for this permission and be able to change the security properties on any object.

Database object permissions

You secure deployment databases by setting permissions on directories and objects (such as J2EE archives). You can restrict access by user or group. Permission settings apply to the directory, all subdirectories (or a specific one), and objects in the directory. Each object or resource within a hierarchy must be secured.

📖 For more information, see “Step 10: Secure your deployment databases” on page 179 in the security checklist.

The following permissions apply to objects in a deployment database.

Permission	Description
Read	For an object, limits who can view the object's design information. For a database or directory, limits who can access child objects. To easily prevent users from accessing objects in a database or directory, deny them Read access at the database or directory level. NOTE: Users who need to run an object in a subdirectory must have Read access to **all** of its parent directories. For more information, see "Making secure application objects executable" on page 171.
Modify	For an object, limits who can change the object. For a database or directory, limits who can add new child objects. To avoid adverse changes to the server, carefully review who has Modify permission to a directory.
Execute (called Default Execute for databases and directories)	Limits who can run application objects (such as J2EE archives) within the selected database or directory. End users typically have Execute permission on objects. Permissions set on an individual application object overrides permissions set on the database or directory.
Set Permissions	Limits who can alter access control information on an object. For example, you may decide that only administrators should be able to change the security properties on an object.

How access works

Different types of server resources have different sets of permissions:

- By default, access to **server administration operations and database objects** is restricted to members of the Administrators group. See "Default server and object security" on page 172.
- Most **directory levels** in the hierarchy allow you to set Read, Write, Execute, and Set Permissions access rights. Each object or resource within a hierarchy can be secured in an identical fashion.
- If access rights are defined at the **database level**, you see all databases but can only expand or open the ones for which you have Read access.
- If you have **Read access to a particular directory**, you can see the named contents of that directory.
- Requests for the contents of the named **object** may be restricted based on permissions set on that object.
- You can change **security access rights** only if you have Set Permissions rights to the deployment database directory or object. You can then use specific access expressions to further limit modifications.
- You can **secure multiple directories** by selecting the Apply To This Directory And All Descendants radio button at the bottom of the permission tabs. You can enable the Require Login For Access check box for selected objects as needed.

📖 For more information, see "Making secure application objects executable" on page 171.

Changing access

You can change object security at the server or cluster level, at the directory level, or at the object level by setting permissions in the SMC.

NOTE: In addition to setting permissions in the SMC, you can also set some types of permissions using SilverCmd SetSecurity, described in the SilverCmd reference chapter of the *Facilities Guide*.

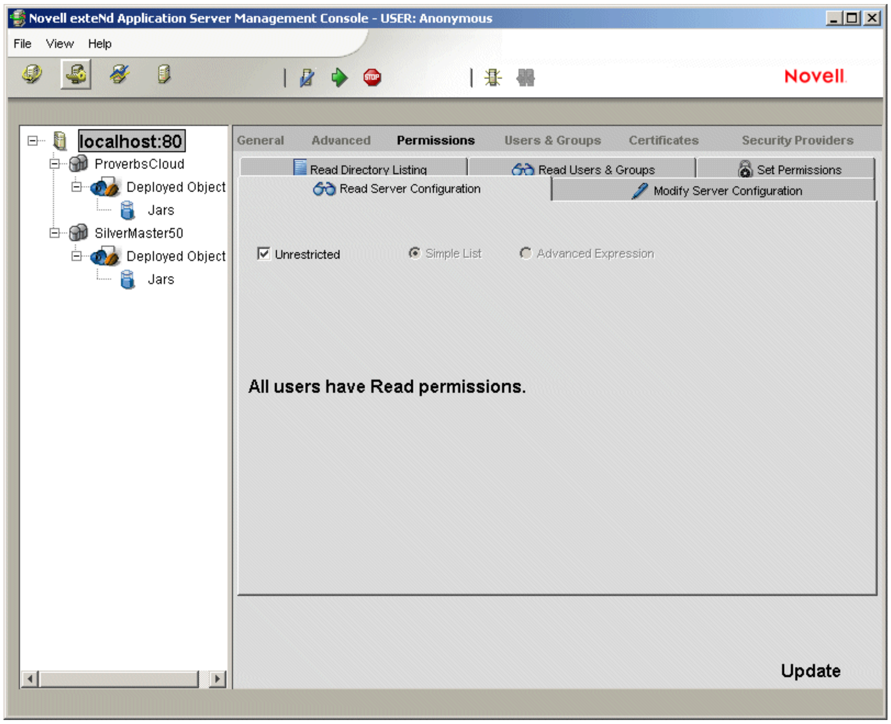
➢ **To change user access:**

1 Start the SMC.

2 Select the **Security** icon from the toolbar.

3 Select the **Permissions** panel.

The SMC view changes. All clusters, servers, and databases being administered by the SMC are now expandable, allowing you to list their contents in order to set permissions at the appropriate level:



☑ Unrestricted ● Simple List ○ Advanced Expression

All users have Read permissions.

4 Select an object or directory and set your permissions as described in "Permission types" on page 166.

Setting permissions on multiple objects You can set permissions on more than one object at a time as long as the objects are in one directory, of the same type, and on the same server: select Shift+Click for contiguous entries or Ctrl+Click for noncontiguous entries and specify the permissions for the selected objects.

Setting the scope of the permissions and whether login is required What radio buttons appear at the bottom of the Permissions panel depend on whether you selected a directory or an object. The following table describes each button option that might appear:

Radio button option	Description	Displays for
Apply to this directory only	Applies permissions only to the selected directory and becomes the default setting for new objects in the directory structure. Does not change permissions for existing objects in the directory.	Directory selections only
Apply to this directory and all descendants	Applies permissions to the selected directory and all existing objects within that directory (including subdirectories) as well as new objects. Overrides any existing settings for existing objects.	
Require login for access	Requires the user accessing the object to be authenticated, either by a client certificate or by being logged on. Sets authentication on a per-object basis. You can also require authentication at the server level. See "Enabling authentication" on page 159.	Object selections only

Restricting permissions

Each tab on the form represents a permission type that applies to the selected directory or object. To restrict access, select a tab, deselect **Unrestricted**, then select either **Simple List** or **Advanced Expression**.

CAUTION: *If you deselect the Unrestricted check box, be sure to specify either a Simple List or an Advanced Expression; otherwise, no one will have access. This problem will also occur if you clear all previous Simple List entries. If you find yourself in the situation where no one has Read access to the server, you can run SilverMasterInit using the -a command-line option. For more information, see "Using the SilverMasterInit program" on page 224.*

Using simple lists

Simple List allows you to specify users and groups that are known to a security provider and have access permission as defined in the provider setup.

➢ **To use simple lists:**

1 Select the **Security** icon from the toolbar, then select the **Permissions** panel.

2 Select the objects or directory you want to set permissions for on the left side of the SMC.

3 In the Permissions panel, select the tab for the permission type you want to restrict.

4 To activate the Simple List button, turn off **Unrestricted** (if selected).

5 Simple List is the default selection. From the form, select the users and groups to whom you want to grant permission, then click >. To select all users or groups, click >>. Selected users and groups are listed in the list box on the right.

NOTE: A local group that contains a global group (that you defined with the NT User Manager) will appear in the SMC with its individual member names listed (instead of the group name). For more information, see "Using NT security" on page 127.

To remove the permission from a user or group, select the user or group and click <.

To remove all users and groups, select <<.

Using advanced expressions

Use **Advanced Expression** to build expressions that let you specify access according to specific criteria. To use advanced expressions, select **Advanced Expression** instead of Simple List (see "To use simple lists:" on page 169). Expressions can include any of the following:

- Identity/group membership
- Built-in functions, which can perform logical and relational operations

➢ To use advanced expressions:

1. Select the objects or directory you want to set permissions for on the left side of the SMC.
2. In the Permissions panel, select the tab for the permission type you want to restrict.
3. If selected, turn off **Unrestricted**; then select **Advanced Expression**.

   The Expression Builder displays selection panes for variables, functions, and operators.

Examples of advanced expressions The following are examples of advanced security expressions:

- Use the day() method to disallow access to the object on weekends (the day() function returns a number from 0 to 6, where 0 indicates Sunday, 1 indicates Monday, and so on):

```
day(now()) >= 1 and day(now())<= 5
```

- Similarly, the following expression disallows access to an object except between 9:00 a.m. and 5:00 p.m. Monday through Friday:

```
(day(now()) >= 1) and
(day(now()) <= 5) and
(hour(now()) >= 9) and
(hour(now()) <= 17)
```

UUID support

The Expression Builder supports *Universally Unique Identifiers* (UUIDs) for security expressions. These provide a more secure system than simple integer IDs. UUIDs are used by default in simple security expressions and are available in advanced expressions.

You can enter UUID expressions directly in the Expression Builder, or choose them from the ID section in the Expression Builder's Functions panel:

Function	Description
userID()	Finds the UUID of the user currently logged on.
userID('*name*')	Finds the UUID of a specific user. To create a qualified name, start with the security realm (SSSW, NT, LDAP, or NIS+), add security authority (such as the NT domain name or LDAP/NIS+ server name), then add the user name. Separate each component with two backslashes (\\). (You need to escape backslashes in user names by doubling the backslashes to four.) ◆ By default, if both the realm and authority are missing, a Silver Security user is assumed. ◆ By default, if the realm is missing, an NT user is assumed, with the first component assumed to be the NT domain. For example: ◆ **userID('bobh')** refers to the Silver Security user bobh (user defined in Silver Security) ◆ **userID('DEVA\\JWilkins')** refers to the NT user JWilkins, who is in the NT domain DEVA

Function	Description
groupID('*name*')	Finds the UUID of a specific group. Group names, like user names, consist of a qualified name. For example: ◆ **groupID('Administrators')** refers to the Silver Security Administrators group ◆ **groupID('DEVA\\Accounting')** refers to the NT Accounting group in the DEVA domain
UUID('*uuidString*')	Finds the UUID corresponding to the string representation of a UUID. This form is used when a group ID or user ID cannot be resolved. The application server translates names into corresponding UUIDs before storing them in the AgAccessRights system table. Similarly, when the client views the expression, the internal form is translated back into human-readable names. If an internal UUID cannot be translated (for example, if it has been deleted), UUID() is used.

Examples The following are examples of security expressions using UUID:

- This example restricts access to the Silver Security users **administrator** and **bobh**:

  ```
  userID() in (userID('administrator'),userID('bobh'))
  ```

- This example restricts access to the Silver Security user **bobh** and the NT user **JWilkins**, who is in the NT domain DEVA:

  ```
  userID() in (userID('bobh'),userID('DEVA\\JWilkins'))
  ```

- This example restricts access to the Silver Security user **administrator** and any user in the Silver Security Developers group:

  ```
  userID() in (userID('administrator')) or userID() userin
  (groupID('Developers'))
  ```

- This example restricts access to the Silver Security user **administrator** and any user in the NT Accounting group, which is in the DEVA domain:

  ```
  userID() in (userID('administrator')) or userID() userin
  (groupID('DEVA\\Accounting'))
  ```

Making secure application objects executable

When you are securing deployment databases, you also need to make sure users can run deployed applications such as EARs, WARs, and EJB JARs.

- J2EE archives are deployed to the Deployed Objects directory of the deployment database, so the deployer needs **Write** permissions to that directory.
- To allow users to run the deployed archives, you usually assign them unrestricted **Execute** permission at the directory level (including all descendants).

➢ **To make secure database objects executable:**

1 Log in to the SMC as an Administrator or a user with Locksmith privilege.

2 Select the **Security** icon from the toolbar.

3 Select **Permissions**.

4 Select the server and database you want to change.

5 Set the initial group permissions as follows:

5a Assign Read and Modify permissions to developers and administrators.

5b Assign Set Permissions access to administrators.

5c Assign unrestricted Execute permission to end users.

IMPORTANT: Select the **Apply to this directory and all descendants** check box on all of the preceding tabs.

6. Click **Update**.
7. Select **Deployed Objects**.
8. Select the **Unrestricted** check box on the **Read** permission tab—but do **not** click **Apply to this directory and all descendants**.
9. Click **Update** to save these settings.
10. Apply the same **Read** permission settings (as in Step 8) to all major directories of your application that you want users to access. While these settings give users access at the parent directory level, they also let you restrict individual objects within those directories.
11. Review individual objects to determine if the **Require login for access** option should be enabled.

Default server and object security

If you choose the installation default, the application server initializes the SilverMaster database. When access to SilverMaster is restricted, all users (except those in the Administrators group) are unable to access administration operations, add databases, and browse directory listings. If your application server is running in a restricted production environment, all users will be required to authenticate themselves when accessing the server.

If you did not choose the **Restrict Access to the Novell exteNd Application Server** installation option, your server resources are not locked down. Installing the application server with **unrestricted** access means that unauthorized users can perform administrative operations and browse directory listings until you lock down access by setting permissions.

📖 See “Ways to lock down a server” on page 173 for other ways to lock down the server.

Default object security

Object security defaults are as follows:

- Users have runtime Read and Execute access to all objects (except administration objects.)
- If you define a particular access at the **directory** level, it becomes the default access for any new objects created within that directory. A directory is a container for objects of one type (like Deployed Objects). When you have selected the Permissions panel in the SMC, you can expand directories to see their contents (assuming you have Read permission on that directory). Any new object takes on the access security of the immediate parent object.

Default group permissions

During installation, SilverMasterInit creates two predefined groups (Administrators and Developers) and sets permissions for both groups. The server requires all users to log in. By default, access to server administration operations is restricted to members of the Silver Security Administrators group for a locked-down server. Access to administration resources (such as who can use the SMC, view session and statistical information, or add and remove users and groups) is only granted to members of this group.

You typically separate who is allowed to read an object from who is allowed to write it. You may also want to define a separate group (such as end users) with Execute permission.

📖 For more information about predefined Silver Security groups, see “About Silver Security users and groups” on page 89.

NOTE: By default, users will have Read access to the top level of the SilverMaster database and directories that will enable them to log in and access any existing deployment databases.

📖 For more information, see “Making secure application objects executable” on page 171.

Locking down servers, clusters, and applications

At some point in the development and deployment process, you will want to *lock down* a deployment database, a server, and/or an entire cluster. Locking down these items helps secure your production and deployment environments by making sure the appropriate access permissions are set.

For example, when deploying an application, you might want to lock down the deployment database so no one can access it except you, then progressively unlock it as appropriate in the production environment.

CAUTION: *If you did not choose the* ***Restrict Access to the Novell exteNd Application Server*** *installation option, your server resources are not locked down. Even if you aren’t ready to lock down and deploy your applications, you should nevertheless* ***lock down*** *your server.*

Ways to lock down a server

The following sections describe different ways you can restrict server access:

Section	Contents
“Default server and object security” on page 172	Describes default application server installation settings
“Using the SMC to lock down the server or an application” on page 173	Describes how to use the SMC to lock down the server or an application
“Using SilverCmd to lock down the server, an application, or a cluster” on page 174	Describes how to use SilverCmd to lock down the server, an application, or a cluster
“Security checklist” on page 174	Provides a list of steps for ensuring that your server is secure
“Excluding robots” on page 180	Describes additional considerations when configuring an application development environment

Using the SMC to lock down the server or an application

➢ **To lock down the specified server:**

1 Restrict the **Read Server Configuration, Modify Server Configuration**, **Read Directory Listing**, **Read User & Groups**, and **Set Permissions** permissions at the server or cluster level to members of the Administrators group.

2 Select the SilverMaster database and restrict **Read**, **Modify**, and **Set Permissions** permissions to members of the Administrators group and apply that restriction to all descendants.

3 Set unrestricted **Read** access to the SilverMaster database to allow users (with appropriate permissions) to access other databases and log in to the server.

📖 For more detailed instructions, see “Step 9: Secure the SilverMaster database” on page 178 in the security checklist.

➢ **To lock down the selected application:**

- Restrict the **Read, Modify**, **Default Execute**, and **Set Permissions** permissions at the database level to members of the Administrators group and apply the restriction to all descendants.

Using SilverCmd to lock down the server, an application, or a cluster

Included with the application server (in the server's **\Samples\SilverCmd** directory) are three sample XML files that are input files for the SilverCmd SetSecurity command. These files show how you can lock down an application, a server, or a cluster using SilverCmd. All three files are well commented with usage notes.

This secure file	Shows you how to lock down	You might want to apply this file
secure_server_sample.xml input file	An entire server	To a server when putting it into production
secure_application_sample.xml input file	An application	To an database just before deploying it in order to secure it properly
secure_cluster_sample.xml input file	A cluster	After you have secured each server in a production cluster using secure_server_sample.xml

📖 For information on using SilverCmd, see the SilverCmd Reference chapter of the *Facilities Guide*.

Securing the production server

The levels and types of security you implement depend on the scale and demands of your particular enterprise. Once your application is ready to deploy (or even if it is already deployed), you should go through each step in the following security checklist to secure your production environment.

If you chose the installation default, the application server restricts access to the SilverMaster database. When access to SilverMaster is restricted, unauthorized users will not be able to access administration operations and browse directory listings.

During application development, you may have opened up access to specific applications and directories. Once your applications are built, it is up to you to make sure that **all** resources are protected from unauthorized access.

Security checklist

TIP: To test your site's security, run any accompanying tests for each step.

Step 1: Design firewalls

Make sure the application server is installed **behind** any firewalls your company uses.

📖 For more information, see "Server configurations" on page 29.

Step 2: Set up a unique database account

Set up a unique database account for the application server to use to connect to each deployment database.

📖 For more information, see Chapter 4, “Data Source Configuration”.

Step 3: (Optional) Set up SSL

If you intend to use SSL communications, obtain a server certificate and install it on the server. You also need to enable the RSA and/or DSA ports and disable HTTP if you want to require only SSL.

📖 For more information, see “Using certificates” on page 136.

Step 4: (Optional) Set up unique ports

For added security, configure separate runtime and administration ports. The server supports ports for the HTTP, HTTPS-RSA, and HTTPS-DSA security protocols. Each unique port you configure excludes URLs and operations that are not associated with it. The separate ports are designed to work in conjunction with your server permission settings.

📖 For more information, see “Setting up separate ports” on page 79.

Step 5: Set up users, groups, and security providers

User and group information can be defined using Silver Security or obtained from an external security system. For Silver Security, all information is stored in the SilverMaster database. For external security, all information is obtained from the external system. In either case, define access to directories and objects.

📖 For more information, see “Managing Silver Security users and groups” on page 90.

Step 6: Require authentication at the server

If you restrict the **Execute** access to all resources in your SilverMaster database, you **must** enable **Require user authentication** at the server level. Otherwise, requiring user authentication is optional.

When a user can access the server without logging in or sending in a certificate, the user is considered Anonymous. It is often a good idea to prohibit Anonymous user access by forcing users to authenticate themselves when they first access the server, through either a certificate or a user ID/password pair. You can also require login on a per-object basis.

➢ To test for unauthorized user access:

- To assess the Anonymous user’s ability to access your site, enter an URL to your application into a browser such as:

  ```
  http://localhost/
  ```

  If the browser displays a **login dialog**, your site is requiring anonymous users to authenticate themselves.

  If the **default page** or a **listing** of your site’s directory contents (or a Read Access Denied message) displays, your site is allowing anonymous access and you may want to complete the following procedure to require user authentication.

➢ To require user authentication:

1. Start the SMC and select a server from the left pane.
2. Select the **Security** icon from the toolbar.
3. Select **General**.
4. Check the **Require user authentication** check box.
5. Click **Update** to save the settings.

📖 For more information, see “Enabling authentication” on page 159.

Step 7: Restrict the directory listing on the server

The application server displays a directory listing when users request certain URLs from a browser. Although a listing of directory entries may not seem critical to your site’s security, you might want to prevent these lists from appearing.

To see whether unauthorized users can access the HTML and non-HTML directory listings, run the following two procedures.

➢ To check whether the HTML directory listing is enabled (and optionally disable it):

1. From your browser, enter the URL of a directory on your server such as:

   ```
   http://localhost/SilverStream/Meta/
   ```

   If your HTML directory listing is **protected**, the following error message displays:

   ```
   You are not allowed to read the specified resource.
   Additional technical details about this error are available.
   ```

   If an HTML directory list displays, you can disable the directory listing access described in the following steps.
2. Open the SMC and select the **Security** icon from the toolbar.
3. Check the **Disable HTML directory listing** check box.
4. Click **Update** to save the settings.

➢ To check whether the non-HTML directory listing is enabled (and optionally disable it):

1. From your browser, enter an URL from a browser such as:

   ```
   http://localhost/SilverStream/Meta/?access-mode=text
   ```

   If your non-HTML directory listing is **protected**, the following error message displays:

   ```
   You are not allowed to read the specified resource.
   Additional technical details about this error are available.
   ```

   If you see the directory contents listed in plain text, you can disable directory listing access as described in the following steps.
2. Open the SMC and select the **Security** icon from the toolbar, then select **Permissions**.
3. Set the **Read Directory Listing permission** to limit access as described in “To check the Read Server Configuration setting of your administration resource:” on page 177.
4. Click **Update** to save the settings.

📖 For more information, see “Enabling authentication” on page 159.

Step 8: Secure the administration resource

The **administration resource** controls your ability to view, modify, and change administrative settings (and permissions to access settings). You secure server objects by restricting permissions on the administration resource. Once you have secured the administration resource, unauthorized users will not be able to perform administrative operations.

Run the following two tests to check access to your server. But even if the test URL does not access protected server information, you should use the SMC to verify that your administration resource security settings properly restrict access to the appropriate users.

➢ To check the general accessibility of your administration resource:

1. Open the SMC.
2. As an anonymous user, try to view and change application objects.

   On a **secure** server you see:

   ```
   Read Access Denied.
   Additional technical details about this error are available.
   ```
3. If you are able to view or change application objects, you should **promptly** run the procedure "To protect administration access:" on page 177.

➢ To check the Read Server Configuration setting of your administration resource:

1. Enter the following URL for your Web site:

   ```
   http://localhost/SilverStream/Administration/
   ```

   On a **secure** server you see:

   ```
   Read Access Denied.
   Additional technical details about this error are available.
   ```

   If you see a listing of site-specific server properties (similar to the following text sample) listed in plain text, your site is **not** protected.

   ```
   com.sssw.loadbalancer.connect.tryInterval=30
   com.sssw.srv.server=exteNd ApplicationServer/n.n
   com.sssw.srv.http.ClientPool.minIdle=0
   com.sssw.loadbalancer.connect.sleepCount=10
   com.sssw.srv.http.ClientPool.minFree=10
   ```
2. If the administration resource at your site is not protected, you should **promptly** run the procedure "To protect administration access:" on page 177.

CAUTION: *If you discover that your administration resource was left unrestricted, you should immediately change the user name and password for each database. These steps will help ensure that anyone who may have accessed this information while the resource was unrestricted can no longer use it.*

➢ To protect administration access:

1. Start the SMC and select a server from the left pane.
2. Select the **Security** icon from the toolbar.
3. Select **Permissions**.
4. On each permission tab, deselect the **Unrestricted** check box to restrict access.

   It is especially important that you restrict the **Read Server Configuration** permission by deselecting the **Unrestricted** check box.
5. Select either **Simple List** or **Advanced Expression**.

6 Select the users and groups or define an expression that specifies who will be assigned the permission on the selected tab. You should limit access on all five tabs (listed in the table in "Administrative server permissions" on page 166) to members of the **Administrator** group (it is given to both Developers and Administrators by default).

🕮 For more information, see "Permission types" on page 166.

Step 9: Secure the SilverMaster database

Special care should be taken when securing the SilverMaster database. In addition to storing resources such as user and group information, SilverMaster stores the login resource and references to all other deployment databases available to the server.

Unless your applications are tightly controlled, you don't typically apply Read restrictions to the SilverMaster database. You must allow Read access to the top level of the SilverMaster database in order for users to be able to access anything on the server. You may want to lock down all resources below the main directory level.

If you accidentally restrict Read access to your SilverMaster database, enable **Require user authentication** on the Server Security panel. If you forget to require authentication, run SilverMasterInit with the -a option to enable user authentication from the command line. You will need to restart the server for the authentication change to take effect.

➢ To test access to your SilverMaster database:

- To see whether your SilverMaster database can be accessed by unauthorized users, follow "Step 10: Secure your deployment databases" on page 179.

  In Step 4 of that procedure, select the SilverMaster database you think you've restricted. Select an object stored below the SilverMaster parent level to test access.

➢ To lock SilverMaster resources below the main directory level:

1 Open the SMC as an administrator and select the **Security** icon from the toolbar.

2 Select **Permissions**.

3 In the left panel, select the SilverMaster database below the desired server.

4 Restrict **Read**, **Modify**, and **Set Permissions** to members of the Administrators group and then click the **Apply to this directory and all descendants** radio button. For more information about permission settings, see the table in "Administrative server permissions" on page 166.

5 Click **Update** to save the settings.

6 Reselect the SilverMaster database and set unrestricted **Read** access (make sure the **Apply to this directory only** is selected).

7 Click **Update** to save the settings.

8 Repeat Step 6 and Step 7 for each of the directories and subdirectories expanded below SilverMaster (such as **Deployed Objects**).

9 Click **Update** to save the settings.

🕮 For more information, see "Configuring deployment databases" on page 46.

Step 10: Secure your deployment databases

You should restrict user access to deployment databases by setting permissions on directories and objects.

Understanding permission settings Permission settings apply to the directory, all subdirectories, and any objects in the directory or any subdirectory. For each application, review every object and specify who is allowed Read, Write, Default Execute, and Set Permission access on each directory level in the hierarchy. Each object or resource within a hierarchy must be secured.

Unless you set restrictions, all users have full access permissions on **objects**. If you define a particular access on a **directory**, it becomes the default access for any new objects created within that directory. When you have selected the Permissions panel in the SMC, you can expand directories to see their contents (assuming you have Read permission on that directory). You can secure multiple directories by selecting the **Apply to this directory and all descendants** radio button at the bottom of the permission tabs. You can enable the **Require login for access** check box for selected objects as needed.

NOTE: If your application objects contain EJBs, you must unrestrict Read access to the parent directory of the objects directory (and to all descendants), since the object directory contains classes that need to be read by the application.

➢ **To test access to database objects (and optionally secure them):**

1. Log in to the SMC as an Administrator or user with Locksmith privilege.
2. Select the **Security** icon from the toolbar.
3. Select **Permissions**.
4. In the left panel, select an object that you think should be restricted that is stored below the parent level of the directory structure.
5. Click each tab and see who has access to this object. Complete the following steps if the permission restrictions are not set correctly.
6. Click each tab and select either **Simple List** or **Advanced Expression**.
7. Select the users and groups or define an expression that specifies who will be assigned the permission on the selected tab.
8. Make sure that someone in the **Administrator** group has **Set Permission** access to this object.
9. Click **Update** to save the settings.

For more information, see “Database object permissions” on page 166 and “Making secure application objects executable” on page 171.

Step 11: Map J2EE security roles

When deploying J2EE archives, map the security roles specified in the deployment descriptor to security principals in the production environment. For example, if the bean developer has defined three security roles, these same roles must be mapped to security groups or individual users on the target server. If you decide to map roles to groups (for ease of maintenance), be sure to verify group membership.

Excluding robots

Robots are programs, such as search engines and Web crawlers, that can traverse many pages on a Web site by recursively retrieving linked pages. The application server supports the commonly used **robot exclusion protocol** that allows you to specify exactly what files and directories robots that conform to the exclusion protocol can access on your Web site.

NOTE: The robot exclusion protocol is purely voluntary. There is no guarantee that a robot conforms to the protocol—but most do.

In this exclusion protocol, the access policy for robots is specified in a file called **robots.txt**. The robots.txt file shipped with the application server tells robots not to proceed below the root of the server (in other words, it disallows all access to all robots that conform to the exclusion protocol). So by default, conforming robots are disallowed all access to your Web site.

You can modify robots.txt to specify the kind of access you want to provide to robots. For example, you might want to allow search-engine access to some directories but not to others.

➢ **To modify robot access:**

- Edit **robots.txt** in the server’s **\Resources** directory.

  📖 For information about the protocol’s format and semantics (including examples), see http://www.robotstxt.org/wc/robots.html.

  The next time a conforming robot requests the access policy from the server, the updated robots.txt file will be read and the information sent back to the robot.

11 Tuning the Server

This chapter describes how to use the SMC to manage server parameters that affect the overall performance and efficient operation of your system. This chapter contains sections on:

- Setting performance parameters
- Managing client connections
- Managing the server content cache
- Managing connection pools
- Managing database connections

Setting performance parameters

To improve performance, you can use the SMC to set buffer sizes as well as server timeout parameters.

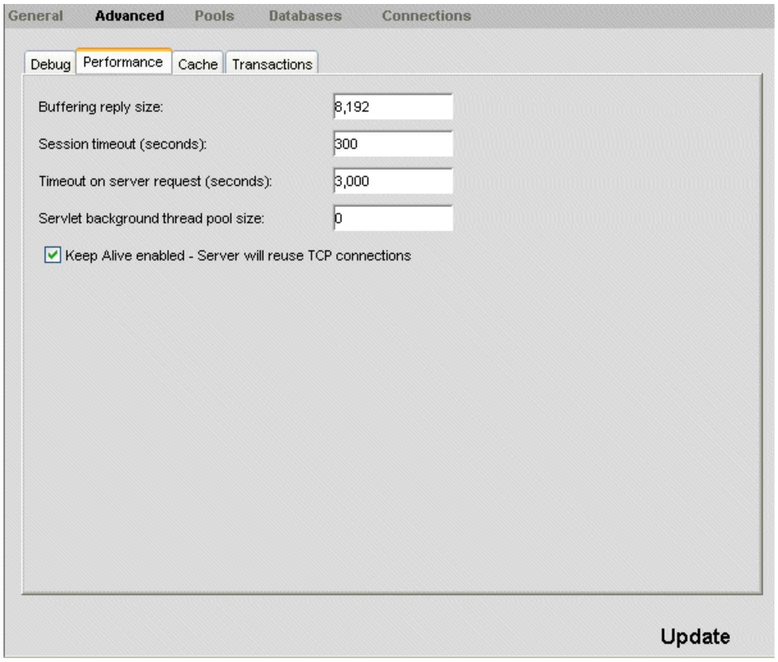
➢ To set performance parameters:

1 Start the SMC.

2 Select the **Configuration** icon from the toolbar.

3 Select **Advanced**.

4 Select the **Performance** tab:



Buffering reply size: 8,192

Session timeout (seconds): 300

Timeout on server request (seconds): 3,000

Servlet background thread pool size: 0

☑ Keep Alive enabled - Server will reuse TCP connections

5 Edit the settings as needed:

Field	Description
Buffering reply size	The size of the packet (in bytes) in which larger replies are gathered, copied, and sent back to the client. Setting this to a larger size might provide better performance—but it is best to limit the size to 8KB or 16KB, depending on available server physical memory and the number of client connections.
Session timeout	The time after which the server will terminate a session with a client after the client goes idle. The default value is five minutes. The session timeout value affects how long session object variables are maintained. The longer the timeout, the longer the memory used for a session will be consumed by the server. In deployed applications that do not use session objects, you probably want to use a short session timeout—but you probably do not want to reduce this value below four minutes.
Timeout on server request	The maximum total server-side time for processing a client request, from the instant the request is received until the reply is sent back. This allows the server to time out in the event of a persistent problem. You should set this value to be substantially longer than the longest expected server response time.
Servlet background thread pool size	Set this value **only** when you have deployed an exteNd Director application containing portlets. When set to a number greater than 0, specifies the number of threads that the portlet container can allocate to render portlets in parallel.
Keep alive enabled	When checked client connections can be reused for additional client requests to the server. The server uses an HTTP(s) listener that accepts client requests via the TCP/IP protocol. The server is configured to have a certain number of threads that handle the client TCP/IP connections (see “Managing client connections” on page 182). ◆ If **Keep alive enabled** is checked (true) after handling a new TCP/IP request, the server reads the next client request from the same client TCP/IP connection. ◆ If **Keep alive enabled** is disabled after handling a new TCP/IP request, the server closes the client TCP/IP request.

6 Click **Update**.

7 To activate the new setting(s), click the **Restart** (server) button.

Managing client connections

This section describes how a client connection is established on the server, and how you can modify connection parameters to improve performance.

NOTE: **Client** connections are not the same as **database** connections. Database connections are described in “Managing database connections” on page 192.

Client sessions and threads

When a client first connects with the server through HTTP, the server establishes the connection by allocating a *thread* (if available) from the connection pool.

About threads

The thread is a lightweight background process that:

- Creates a server session
- Executes the HTTP request by hooking up to the appropriate server components and passing the result back to the client

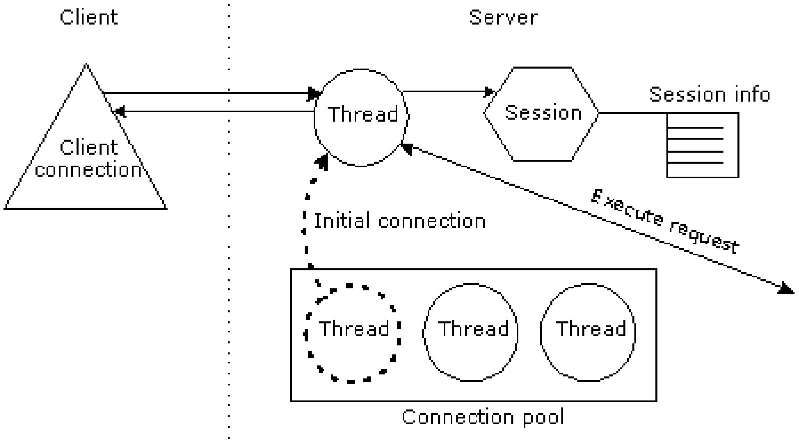
The thread is associated with the client connection until either the client or the server closes it. The server will close the connection according to the settable connection parameters. When the connection is closed, the thread is returned to the client connection pool.

About session objects

The *session* is an object on the server that does the following:

- Stores information about the client (such as user ID, login, and hostname).
- Stores the application data included programmatically in the client's session object (such as in a shopping cart application, which needs to store purchases and other information for each client along the way).
- Remains alive for a specified time (session timeout period) after all threads have been returned to the pool. (In certain cases there can be more than one client thread associated with a session.) This allows the client to reconnect to the session within the specified time. The default timeout is five minutes. Once a session object has been deleted by the server, its contents are gone.

The following diagram shows a client connection with the server session and its associated thread:



Client connection parameters

The application server imposes a limit on the total number of allowable client connections (threads). Within this allowance, you can modify connection parameters in order to manage the server load in your production environment. The server defines client connections in terms of individual connection states and overall loads.

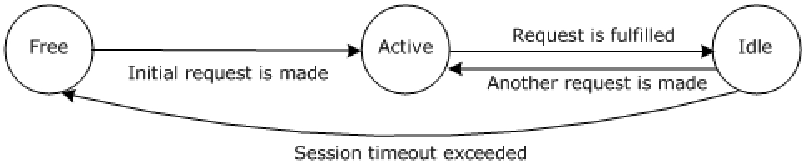
Connection states

Each client connection (thread) exists in one of the following states:

- **Free**, if it is not in use
- **Active**, if it is connected and in use—and a request is in progress
- **Idle**, if it is connected and in use—but no request is in progress on the connection

Once a thread has been idle for the session timeout period, the application server closes the connection and returns the thread to the connection pool. (You can configure the timeout period. See “Setting performance parameters” on page 181.)

A thread typically has the following life cycle:



Load levels

The server has two load levels: **light** and **busy**. It behaves differently when requests for new connections are made, depending on its load level.

How the application server operates based on load levels Here is what happens when the server receives a request for a new connection:

1. A request is made for a new connection to the server.
2. The application server responds based on its load level:

Load level	What the application server does
Light	The new request is allocated a free thread. When an active connection goes idle, it is kept open until it times out.
Busy	The server closes older idle connections (enough to bring the server to the **light** load level if possible), returning the threads to the connection pool, then allocates a free thread to the new request. If there are no idle connections (that is, all the allowable number of connections are active), the new request is refused—unless you configure the server to allow a dynamically allocated client connection (described in “Setting the connection parameters” on page 185).

Specifying how the load level is determined You can specify how the server determines whether the load is **light** or **busy** using the SMC. In the descriptions that follow, the terms in **bold** refer to the label for the properties in the SMC:

- **Maximum number of client connections**. Determines the total number of allowable client connections.
- **Idle client connection reserve**. Determines the server load level:
  - The server load level is **busy** when the number of free connections is **lower** than the number of **Free client connections for light**.
  - The server load level is **light** when the number of Idle client connections is **greater** than the number of **Idle client connections reserve**.

Setting the connection parameters

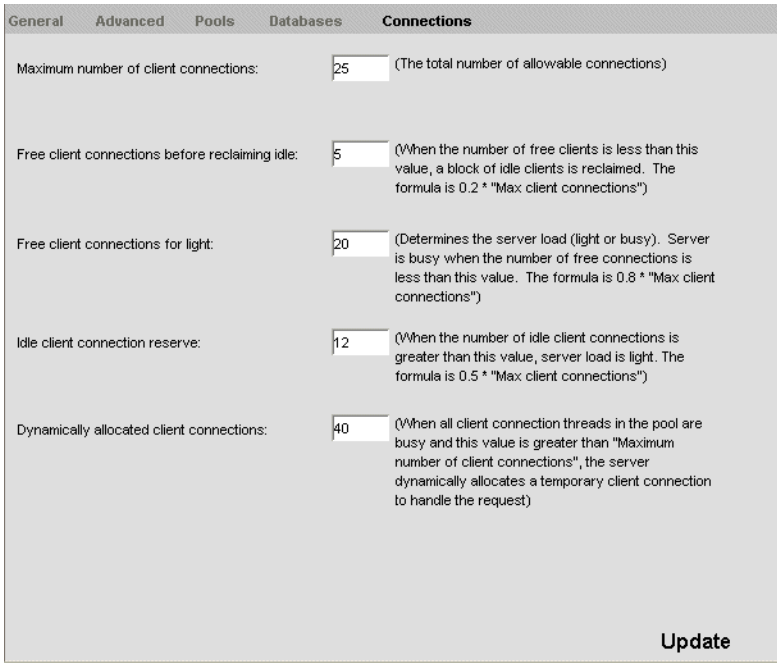
You can set the connection parameters in the SMC.

➢ **To set client connection parameters:**

1 Start the SMC.

2 Select the **Configuration** icon from the toolbar.

3 Select **Connections**:



Maximum number of client connections: 25 (The total number of allowable connections)

Free client connections before reclaiming idle: 5 (When the number of free clients is less than this value, a block of idle clients is reclaimed. The formula is 0.2 * "Max client connections")

Free client connections for light: 20 (Determines the server load (light or busy). Server is busy when the number of free connections is less than this value. The formula is 0.8 * "Max client connections")

Idle client connection reserve: 12 (When the number of idle client connections is greater than this value, server load is light. The formula is 0.5 * "Max client connections")

Dynamically allocated client connections: 40 (When all client connection threads in the pool are busy and this value is greater than "Maximum number of client connections", the server dynamically allocates a temporary client connection to handle the request)

4 Specify settings as follows:

Field	Description and Guidelines
Maximum number of client connections	Determines the maximum number of connections in the client connection pool. This is a hard limit. **Guideline**—Set this value to: `Maximum number of simultaneous sessions + 10 %` (Assuming there will be about one active connection per session.) Setting this parameter too high will result in excessive consumption of system resources by the application server. Setting it too low can result in connections being refused. For times when peak server loads exceed the size of the client connection pool and you do not want to drop incoming client requests, you can also set the server property Dynamically Allocated Client Connections (described below). **Default values for non-NetWare platform installs:** ◆ Express or Server Express—25 ◆ Server or Custom—250 **Default values for NetWare installs:** ◆ exteNd J2EE Web Application Server—250 ◆ Customized NetWare Server—25

Field	Description and Guidelines
Free client connections before reclaiming idle	When the number of free connections is lower than this value, the application server reclaims a block of idle clients. **Guideline**—Set this value to: `0.2 * Maximum number of client connections` **Default values for non-NetWare platform installs:** ◆ Express or Server Express—5 ◆ Server or Custom—50 **Default values for NetWare installs:** ◆ exteNd J2EE Web Application Server—50 ◆ Customized NetWare Server—5
Free client connections for light	Determines application server load level. Application server load level is busy when the number of free connections is less than this value. **Guideline**—Set this value to: `0.8 * Maximum number of client connections` **Default values for non-NetWare platform installs:** ◆ Express or Server Express—20 ◆ Server or Custom—200 **Default values for NetWare installs:** ◆ exteNd J2EE Web Application Server—200 ◆ Customized NetWare Server—20
Idle client connection reserve	When the number of idle client connections is greater than this value, application server load is light. Set this value low if your application has a stable user load, such as with an intranet application. Setting it low means that the application is usually in a light load, so idle connections are maintained longer—allowing existing users to maintain their connections longer. For Internet applications (where the user load can vary dramatically), you might want to set it relatively high—to better ensure that new users can get a connection. Idle connections are terminated more quickly. **Guideline**—Set this value to: `0.5*Maximum client connections` **Default values for non-NetWare platform installs:** ◆ Express or Server Express—12 ◆ Server or Custom—125 **Default values for NetWare installs:** ◆ exteNd J2EE Web Application Server—125 ◆ Customized NetWare Server—12

Field	Description and Guidelines
Dynamically allocated client connections	Allows the application server to create client threads dynamically. These dynamically allocated threads provide a cushion for times when peak application server loads exceed the size of the client connection pool. They do not use up system resources during off-peak periods. **Guidelines**: ◆ **Not setting this value**—Set the Maximum Number of Client Connections so that there are enough threads available during peak usage periods; otherwise, some connection requests will be dropped. During off-peak periods, these extra threads use application server resources. ◆ **Setting this value**—Set the Maximum Number of Client Connections value to something less than the peak size—slightly larger than the normal number of simultaneous requests handled during off-peak hours—and set the Dynamically Allocated Client Connections to handle the peak usage. This means that most of the time the pool can handle all requests without requiring additional thread allocation, and threads will be dynamically allocated to handle traffic peaks. The application server is still protected against denial-of-service flooding attacks by the Dynamically Allocated Client Connections, because the application server will still drop incoming requests if this value is exceeded. **Default values for non-NetWare platform installs:** ◆ Express or Server Express—40 ◆ Server or Custom—250 **Default values for NetWare installs:** ◆ exteNd J2EE Web Application Server—500 ◆ Customized NetWare Server—250

5 Click **Update**.

The new settings take place immediately.

Managing the server content cache

The application server uses server-side caching for various purposes, including resource naming and attribute information. The application server also stores (or caches) the contents of file resources in memory or on disk. Caching is mostly invisible to users except that it affects performance.

The two caches

There are two separate caches for every application server:

Cache	Description
Memory cache	This cache is held completely in memory and is intended for holding smaller files
Disk cache	This cache is on disk and is intended for larger files

What you can configure

Each cache has two settings you can configure:

Setting	Description
The maximum content size	The maximum size of any individual file allowed into the cache
The maximum cache size	The maximum size of the cache itself (the total size of all the files in the cache)

You can also specify the directory used for disk caching.

How the application server uses the caches

The application server uses the most appropriate cache for each file based on the file’s size. Small files are stored in the memory cache, and larger files are stored in the disk cache. Very large files are not cached at all. The server uses an LRU (least recently used) algorithm within each cache, throwing out old (not recently used) files when the cache becomes full.

➢ **To set cache settings:**

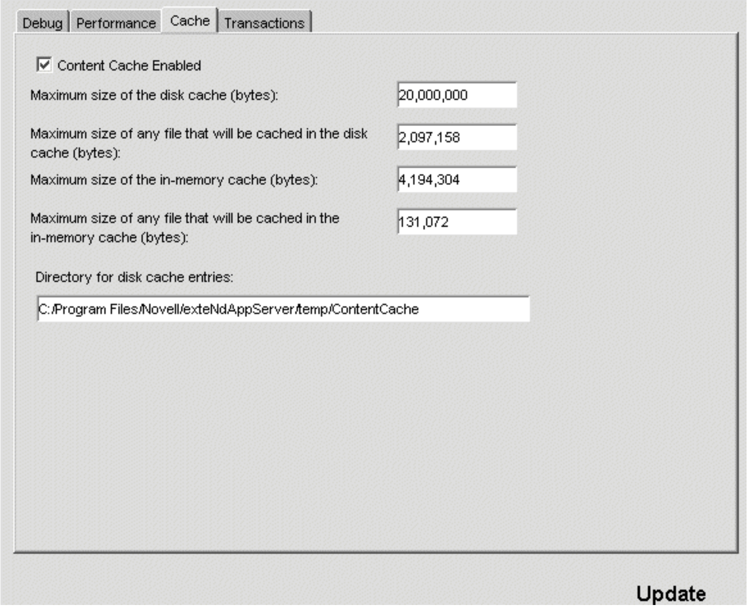
1 Start the SMC.

2 Select the **Configuration** icon from the toolbar.

3 Select the **Advanced** option.

4 Select the **Cache** tab.

The following panel displays:



☑ Content Cache Enabled

Maximum size of the disk cache (bytes): 20,000,000

Maximum size of any file that will be cached in the disk cache (bytes): 2,097,158

Maximum size of the in-memory cache (bytes): 4,194,304

Maximum size of any file that will be cached in the in-memory cache (bytes): 131,072

Directory for disk cache entries:

C:/Program Files/Novell/exteNdAppServer/temp/ContentCache

5 Specify settings as follows:

Field	Description
Content Cache Enabled	Enables or disables the content cache. If unchecked, all content caching is turned off. Caching of frequently requested file resources will improve the response time for these resources. Turn off only for debugging purposes.
Maximum size of the disk cache	The maximum size of the disk cache, in bytes. The total size of all files in the cache is always less than or equal to this. If set to 0, the disk cache is disabled. In general, make this cache large enough to store the remainder of frequently used files after you have determined the in-memory cache size (see the description of "Maximum size of the in-memory cache" below).
Maximum size of any file that will be cached in the disk cache	The maximum size of any file that is cached in the disk cache, in bytes. Files less than or equal to this size but too large for the in-memory cache are cached in the disk cache. Files greater than this size are not cached.
Maximum size of the in-memory cache	The maximum size of the in-memory cache, in bytes. The total size of all files in the cache is always less than or equal to this. If set to 0, the in-memory cache is effectively disabled. In general, make this size as large as possible without causing excessive paging activity on your system.
Maximum size of any file that will be cached in the in-memory cache	The maximum size of any file that is cached in the in-memory cache, in bytes. Files less than or equal to this size will be cached in the in-memory cache.
Directory for disk cache entries	The directory in which the disk cache entries are stored. By default, this is a subdirectory of the application server's installation directory. The server tries to create this directory if it doesn't exist. The server also tries to clear all the cache files from this directory when it starts up. (You might want to have a directory that is dedicated to the cache.) If the server cannot find or create this directory, the disk cache is disabled.

6 Click **Update**.

The new settings take effect immediately.

Managing connection pools

This section contains the following topics:

- About connection pool connections
- Setting the number of connection pool connections

About connection pool connections

The application server handles all connections between a J2EE application and a data source via a connection pool. The connections in the connection pool are typically TCP/IP connections, established between the application server and the data source through JDBC or a resource adapter archive (RAR).

Most database or enterprise information system (EIS) servers place a limit on the number of connections that can be open simultaneously. You can use the SMC to set the minimum (initial) and maximum number of connections used by the application server for each data source.

For maximum performance, the maximum number of open data source connections you set will normally equal the maximum number of users who will be simultaneously querying or updating the data source.

Determining the correct number of connections

Setting a maximum that is too small will result in degraded performance: if a client attempts to access the data source when all connections are in use, the client will be blocked until an in-progress query or update is completed. You may want to experiment with different maximum settings to optimize your server performance.

If other applications will be establishing connections to the data source server independently of the application server, you may need to reduce the maximum number of open application server connections to make sure data source connections are available for all processes that require access to the data source.

Consult your DBMS or EIS documentation and make sure your data source has been configured to accept the number of client connections specified in the SMC, taking into account other applications that might be accessing the data source.

Setting the number of connection pool connections

You can use the SMC's **Pools** tab to manage both JDBC and Connector connection pools, including:

- Minimum and maximum number of connections
- Idle timeouts
- Connection timeouts
- Logging levels

You specify these values for each individual connection pool.

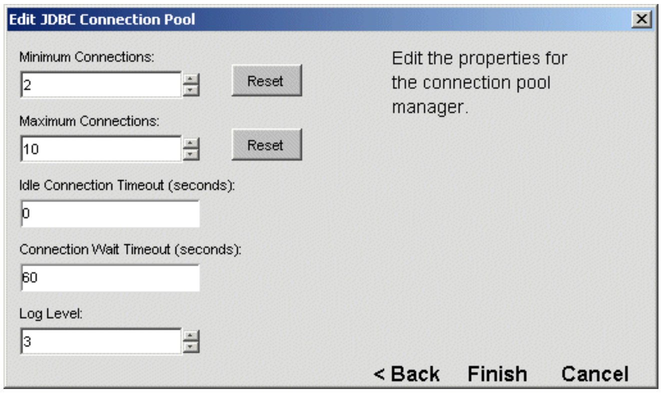
➢ **To update connection pool settings:**

1. Start the SMC.
2. Select the **Configuration** icon from the toolbar.
3. Select **Pools**.
4. Select either **Connector** or **JDBC** to display the available connection pools.
5. Highlight the connection pool whose connection values you want to modify, and choose **Edit**.

   The JDBC Edit Connection Pool Wizard displays.

   You can review various settings for the connection pool, but you can update only the minimum and maximum connection values and the connection pool timeout values.

6 Choose **Next** to proceed through the wizard until you reach the panel that displays the minimum and maximum connection values shown here:



Edit JDBC Connection Pool

Minimum Connections: 2

Maximum Connections: 10

Idle Connection Timeout (seconds): 0

Connection Wait Timeout (seconds): 60

Log Level: 3

Edit the properties for the connection pool manager.

Field	Description
Minimum Connections	The default minimum number of connections for a connection pool. The application server will immediately establish this many connections when it is started and keep them open as long as the server is running. NOTE: If the data source crashes and then restarts, the application server will drop the old connections and reestablish new ones as needed. It will not immediately establish what had been specified as the minimum number of connections.
Maximum Connections	The default maximum number of connections for the connection pool. The application server will open connections on demand up to this number. Make sure the maximum number of open application server connections is less than or equal to the total number allowed by the EIS; otherwise, the application server will get errors from the data source server when it attempts to open a connection.
Idle Connection Timeout	The idle timeout, in seconds. The default is 60 seconds. When set to -1, idle timeout is disabled and no idle connections are ever closed.
Connection Wait Timeout	The connection wait timeout, in seconds. The default is 30 seconds. When set to -1, clients are forced to wait until a connection becomes available.
Log Level	The levels are: 0—Logging is turned off 1—Logs basic connection pool operations 2—Level 1 with more detailed operations and error messages 3—Level 2 with exception stack traces and trace output produced by JDBC driver or Connector resource adapter Messages are written to the server console.

7 Click **Update**.

8 To activate the new settings, click the **Restart** button.

Managing database connections

This section contains the following topics:

- About database connections and performance
- Setting the maximum and minimum number of database connections

About database connections and performance

The application server handles all connections between a client and a deployment database. Each database connection is typically a TCP/IP connection, established between the application server and the database server through JDBC.

Most database servers place a limit on the number of database connections that can be open simultaneously. You can use the SMC to set the minimum (initial) and maximum number of connections used by the application server for each database.

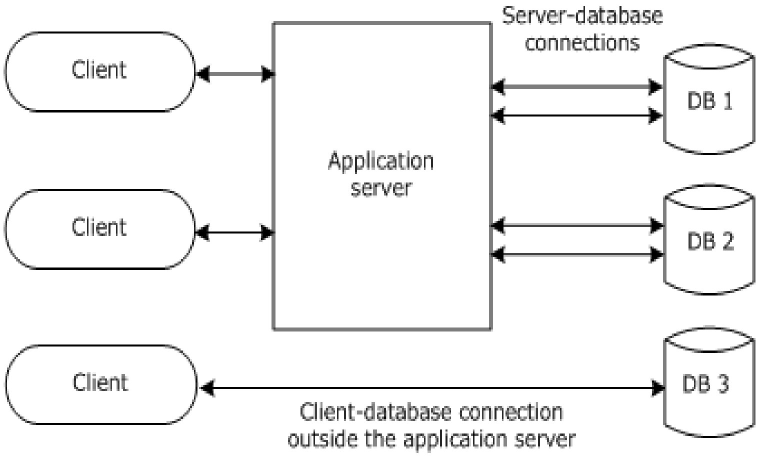
The maximum number of open deployment database connections you set will normally equal the maximum number of users who will be simultaneously querying or updating the database.

Determining the correct maximum number of connections

Setting a maximum that is too small will result in degraded performance: if a client attempts to access the database when all connections are in use, the client will be blocked until an in-progress query or update is completed. You may want to experiment with different maximum settings to optimize your server performance.

If other database users will be establishing connections to the database server independently of the application server, you may need to reduce the maximum number of open connections. This ensures that database connections are available for non–application server users.

You should consult your DBMS documentation and make sure your database has been configured to accept the number of client connections specified in the SMC, taking into account other applications that may be accessing the database:


Server-database connections
Client
Client
Client
Application server
DB 1
DB 2
DB 3
Client-database connection outside the application server

Setting the maximum and minimum number of database connections

You can set the default maximum and minimum number of database connections, as well as the number of connections for specific databases. You do this on the Databases panel in the SMC.

Setting the default number of connections

You can set the default minimum and maximum number of database connections. These values are used by:

- Subsequently added databases
- All databases currently on the server that are using the default minimum or maximum number of connections (databases whose minimum or maximum number of connections have not been changed from the default)

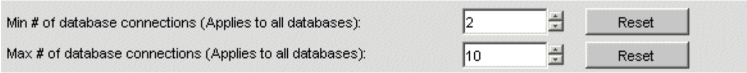
NOTE: To override the defaults for a specific database, see "Setting the number of connections for a specific database" on page 194.

➢ To set the default maximum and minimum number of database connections:

1 Start the SMC.

2 Select the **Configuration** icon from the toolbar.

3 Select **Databases**:



Min # of database connections (Applies to all databases): 2

Max # of database connections (Applies to all databases): 10

4 Edit the fields at the top of the panel as needed.

Field	Description
Min # of database connections (Applies to all databases)	The default minimum number of connections for a database. The application server will immediately establish this many connections when it is started and keep them open as long as the server is running. NOTE: If the database crashes and then restarts, the application server will drop the old connections and reestablish new ones as needed. It will not immediately establish what had been specified as the minimum number of connections.
Max # of database connections (Applies to all databases)	The default maximum number of connections for a database. The application server will open connections on demand up to this number. Make sure the maximum number of open connections is less than or equal to the total number allowed by the database server; otherwise, the application server will get errors from the database server when it attempts to open a connection. 📖 For more information, see "About database connections and performance" on page 192.

5 Click **Update**.

Resetting the defaults

You can reset the default minimum and maximum number of connections. This applies to **all** databases on the server.

➢ To reset the defaults:

- Click **Reset** next to Min # or Max # of database connections at the top of the Databases panel.

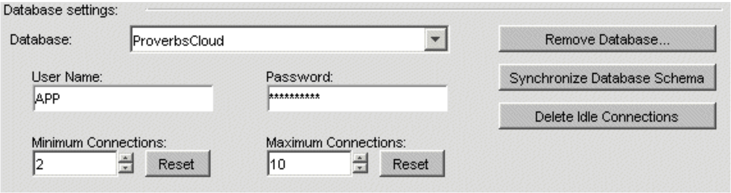
  The default minimum or maximum number of connections is reset, and **all** databases on the server are reset to use the new default values.

Setting the number of connections for a specific database

You can override the default numbers of database connections for a specific database.

➢ To override the default minimum/maximum connections for a specific database:

1. Start the SMC.
2. Select the **Configuration** icon from the toolbar.
3. Select **Databases**.
4. Select the database you want to configure from the list box (under Database settings):



Database settings:

Database: ProverbsCloud

User Name: APP

Password: **********

Minimum Connections: 2

Maximum Connections: 10

5. Set the minimum and maximum number of database connections for the selected database by editing the fields below the database name:

Field	Description
Minimum Connections	The application server will immediately establish this many connections when it is started and keep them open as long as the server is running. Setting a value here overrides the default value. NOTE: If the database crashes and then restarts, the application server will drop the old connections and reestablish new ones as needed. It will not immediately establish what had been specified as the minimum number of connections.
Maximum Connections	The application server will open additional connections on demand up to this number. Setting a value here overrides the default value. Make sure the maximum number of open connections is less than or equal to the total number allowed by the database server; otherwise, the application server will get errors from the database server when it attempts to open a connection. For more information, see "About database connections and performance" on page 192.

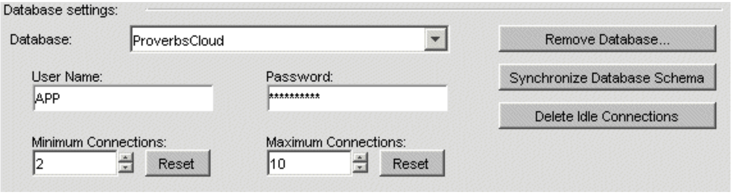
6. Click **Update**.

Restoring the values to the defaults

You can reset the minimum or maximum number of connections for a specific database to be the default values.

➢ To restore minimum/maximum connections to the defaults:

1. Select the database you want to configure from the list box.
2. Click **Reset** next to Minimum Connections or Maximum Connections for the selected database:



Database settings:

Database: ProverbsCloud

User Name: APP

Password: **********

Minimum Connections: 2

Maximum Connections: 10

The minimum or maximum number of connections is reset to be the default value (shown at the top of the panel).

12 Administering a Cluster

This chapter describes how the Novell exteNd Application Server uses server clustering to implement load balancing and failover. It explains how to set up and maintain a clustered environment. The chapter contains the following sections:

- Server clustering
- Cluster components
- Component failover
- Setting up a server cluster
- Administering a server cluster
- Specifying a server's relative load weight
- Managing component failover
- Dissolving a cluster
- Changing the clustering components' properties
- Installing certificates in a cluster

Server clustering

In order to accommodate high processing loads, the application server implements load balancing using server clustering.

What is a server cluster?

A *server cluster* is a set of servers running on different hosts that share the processing load. In the application server environment, a cluster is a group of independent systems working together as a single system connected to the same SilverMaster database. In this configuration, clients interact with a cluster as though that cluster were a single high-performance, highly reliable application server.

NOTE: A cluster of servers can handle demand over time more efficiently than one large machine can, because the sum of the bandwidth and resources of multiple machines is greater (and less expensive) than that of a single large machine.

Benefits of server clustering

Server clustering provides the following benefits:

Benefit	Description
Scalability and better performance	You can increase the number of requests that get processed over a period of time. When the overall load exceeds the capabilities of the systems in the cluster, additional systems can easily be added to the cluster.
Cache management	If a change is made to an application, cache management ensures that the change is propagated to each server in the cluster.

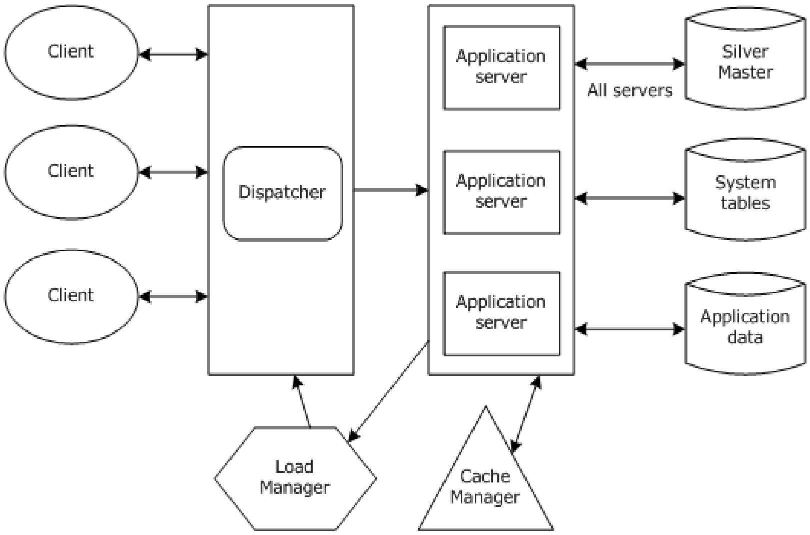
Benefit	Description
Load management	Client requests are distributed to servers in order to optimize overall performance. You can also use a third-party load manager.
Failover	Failover capability is provided for each component of the server cluster. Specific system failover capabilities are described in “Managing component failover” on page 212.

Cluster components

An application server cluster includes these components.

Component	Number	Purpose
SilverMaster database	One	Keeps track of cluster membership
Application server	One or more (usually at least two)	Serves applications
Load Manager	One (optional but requires Dispatcher when used)	Manages the activity of the Dispatcher
Dispatcher	One (optional but requires Load Manager when used)	Used by the Load Manager to redirect requests to application servers
Cache Manager	One	Keeps server caches in sync

The cluster components are configured as follows:



Cluster requirements A cluster configuration must meet these general requirements:

- Each server in the cluster must communicate with the same SilverMaster catalog.
- Each server must have a unique address and port.
- Each component in the cluster must be able to communicate over TCP/IP.
- A client machine capable of running the SMC must be available to create and configure the cluster.

Cluster options You have the following options within the cluster configuration:

- Servers can reside on any machine in the network.
- You can have an unlimited number of servers per cluster.
- Servers and components can run on different platforms.

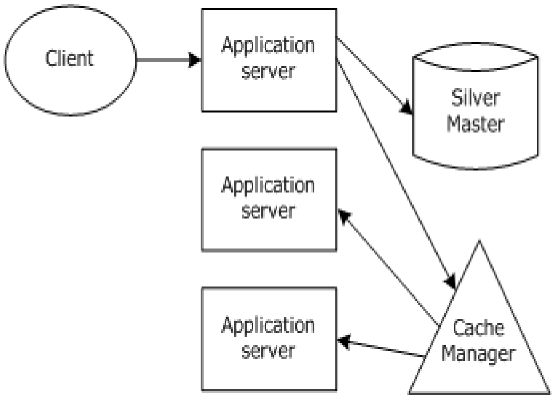
The Cache Manager

Each application server has an intelligent caching mechanism for storing commonly accessed data in memory (such as security information). Reading this information from the database for each request would be very inefficient, so the application server maintains this information in cache memory. Whenever the information is updated on the server (for example, a new security permission is applied or a J2EE archive is deployed), the application server's cache is updated as well.

Maintaining this cache across multiple servers in a cluster is important for application and data consistency. In a clustered environment, multiple servers can simultaneously change the same data in the application server's system tables, and this situation might leave data cached in memory in a corrupted or inconsistent state. The Cache Manager is responsible for preventing such conflicts by ensuring that other servers in the cluster are notified when any server invalidates cache objects. All servers then discard their invalid cache entries and get an updated version of the resource the next time the object is needed.

How the Cache Manager works

The Cache Manager can run on a separate machine or on any machine in the server cluster. The Cache Manager must be up and running before any servers in a cluster start. At startup, the application server reads the clustering information about the setup from the SilverMaster catalog and initiates contact with the Cache Manager. The Cache Manager registers the server's existence and identifies it as a member of the cluster:



If a server modifies an existing object, it notifies the Cache Manager that the specific object must be invalidated on other servers. Since all objects in the application server environment can be denoted by an URL, the server calls the Cache Manager to invalidate the specified URL for all servers. The Cache Manager calls each registered server (except the one that initiated the invalidation) and tells it to invalidate the object identified by the URL.

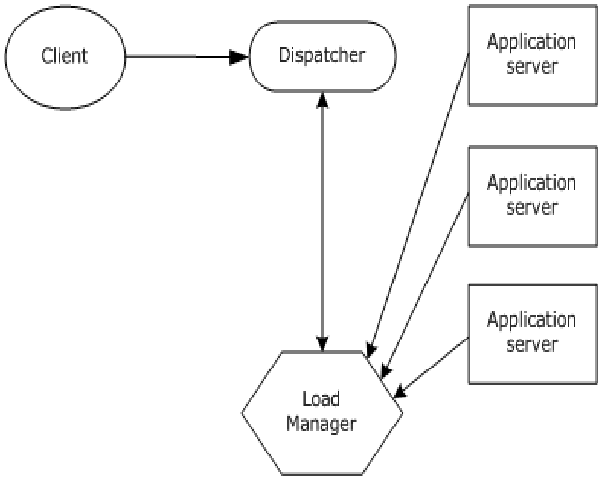
The Load Manager

The Load Manager is a program that can run as a service or regular process on any machine in the same network as the cluster. The Load Manager's role is to keep track of each active server and its relative processing *load* (or weight, as explained in "Distribution mapping" below) in the cluster. Based on this information, the Load Manager generates a distribution map, which it transmits to the Dispatcher in the configuration.

How the Load Manager works

The Load Manager must be up and running before any servers in a cluster start. At startup, the server reads the clustering information about the setup from SilverMaster and initiates contact with the Load Manager to register the server's existence. The Load Manager first registers that the server is up, then determines how to contact the server and factors the server into the distribution map. At Load Manager startup, the Load Manager reads the Dispatcher information about the setup from its property file and initiates contact with the Dispatcher so that it can transmit the distribution map in the future.

The figure below shows the Load Manager:



Distribution mapping

The distribution map, which is dynamically generated by the Load Manager, determines the processing load for each server in the cluster. The distribution weight for each server is represented by an integer between 1 and 10.

By default, a *round-robin system* is used to distribute the load, which means that all servers in the cluster get an equal number of hits over time and have the same weight.

You can modify this distribution. For example, if you set server 1 at 8, server 2 at 4, and server 3 at 2, over a period of time server 1 would get twice as many hits as server 2, and server 2 would get twice as many hits as server 3. The table that follows shows other examples of weight settings:

Servers in clusters	Weight	Result
1, 2, 3	null	Round robin
1, 2, 3	1, 1, 1	Round robin
1, 2, 3	2, 4, 6	Server 1 is least hit, server 2 eventually has twice as many hits as server 1, and server 3 eventually has three times as many hits as server 1
1, 2, 3	1, 1, 10	Server 3 eventually has 10 times as many hits as servers 1 and 2

📖 For information on modifying the distribution, see "Specifying a server's relative load weight" on page 212.

The Dispatcher

The Dispatcher serves as the entry point for Web clients into a server cluster: clients initially access the cluster through the Dispatcher's URL.

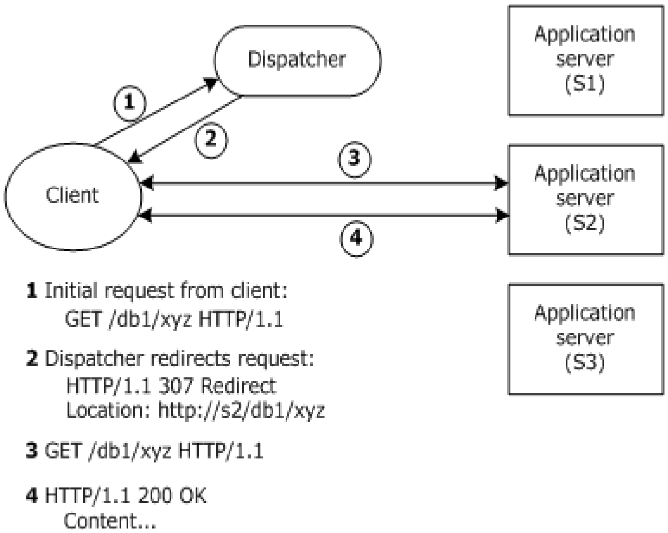
The Dispatcher is a lightweight HTTP/HTTPS-supported program that communicates with other cluster components to dispatch client requests to servers according to the distribution map the Load Manager provides. The Dispatcher can be run as a service or as a regular process on any machine in the same network as the cluster servers. The Dispatcher and Load Manager can run on separate machines and on separate platforms, and do not need access to the SilverMaster database.

How the Dispatcher works

The first time the Dispatcher is called from the client, it finds the best server for the request, based on the distribution map it receives from the Load Manager. Then, using a standard process called HTTP redirect, it redirects the request back to the client. The client then sends the request directly to the designated server.

Once a session has gone through a Dispatcher to a particular server, all client communication is sent directly to the server, not through the Dispatcher; and the redirection is not masked (meaning that in the browser the user sees the URL for the server, not the URL for the Dispatcher).

The following figure shows the sequence in the HTTP redirect process:



1 Initial request from client:
GET /db1/xyz HTTP/1.1

2 Dispatcher redirects request:
HTTP/1.1 307 Redirect
Location: http://s2/db1/xyz

3 GET /db1/xyz HTTP/1.1

4 HTTP/1.1 200 OK
Content...

The Dispatcher supports HTTPS (as well as both HTTP 1.0 and HTTP 1.1). This means you can install server certificates after you set up a cluster. See "Administering a server cluster" on page 209.

Using third-party dispatching solutions

While the application server's software Dispatcher is an excellent solution for many load-balancing and failover needs, you may require functionality that it does not provide.

You may want to use a third-party dispatching solution for the following reasons:

Reason	More information
Complex load-balancing algorithm	When you want more control over how load distribution is calculated, you may want to use another dispatching solution.
DNS masking	Since the software Dispatcher performs a simple HTTP redirection to an available server in the cluster, the browser must be able to establish a direct connection to that server. All clients therefore must be able to resolve all TCP/IP host names for server members. On the other hand, dispatching solutions with DNS masking capabilities process all incoming and outgoing traffic—and thus all servers can be resolved using one common host name. This capability is particularly useful in Internet applications where it is desirable for users to see only one host name (such as www.company.com)—instead of a host name for each server and dispatcher.
Session-level failover	For J2EE applications whose deployment plans indicate failover support (WARs marked as distributable and EJB JARs marked as recoverable), many third-party dispatchers can automatically reroute a session to another server in the cluster transparently to the user. For session-level failover, rerouting is required.
Global dispatching	You may want to have dispatchers that can do more complex routing—for example, to different sites around the world.

In situations like these, you may want to substitute a third-party dispatching solution for the Load Manager and Dispatcher. You would still use the application server's Cache Manager to maintain a consistent cache and would still use the SMC to manage the server.

Any standard HTTP server load-balancing solution should work with the application server.

Component failover

The application server provides system failover and recovery in the event of transient failure and persistent failure. If any component in a cluster fails, the failure is detected by a background program called **SilverMonitor**.

About SilverMonitor

SilverMonitor observes the state of the daemons, processes, and services running on the system. SilverMonitor monitors each component of the server cluster; if any component fails, SilverMonitor detects the failure and attempts to restart the component.

SilverMonitor usually ensures that failed components recover quickly. If a failure is persistent (due, for example, to a hardware failure), SilverMonitor gives up after a predefined number of attempts in order to conserve system resources.

NOTE: SilverMonitor runs in each cluster by default. It is also provided as a server startup option through which you can define certain program parameters. For more information, see "Using SilverMonitor" on page 223.

If SilverMonitor cannot restart a server

When a server in a cluster goes down and is not restarted promptly, the following sequence occurs:

1. The Cache Manager detects the failure.
2. The Load Manager detects the failure and removes the server from the distribution map.
3. The Load Manager sends an updated map to the Dispatcher so that new clients are not redirected to the failed server.

When the server is restarted

When the server is restarted, the following sequence occurs:

1. The server reconnects to the Cache Manager.
2. The server reconnects to the Load Manager.
3. The Load Manager rebuilds the distribution map and sends it to the Dispatcher.
4. The server begins to receive client requests.

When a server fails, connected clients lose their connections. They need to either restart or send another request from the browser to the Dispatcher.

Persistent failure

If restarting a server takes a significant amount of time, the cluster components force a reconfiguration of the cluster so that incoming client requests can be serviced as efficiently as possible. During a persistent failure, the active components respond as follows:

When this component goes down	This happens
Application server	The failure is detected by the Load Manager, which instructs the Dispatcher to redirect traffic to active servers, as described above.
Load Manager	The distribution map cannot be updated, but the Dispatcher still works using the cached version of the map.
Dispatcher	No new connections can be made. Existing sessions are not affected.
Cache Manager	Applications whose logic or properties have not changed continue to run correctly.

Setting up a server cluster

The application server’s clustering components (Cache Manager, Load Manager, and Dispatcher) are available with certain editions of the application server.

➢ **To set up a server cluster:**

1. Install the first server and create the SilverMaster. If you have already completed the installation, make sure the SilverMaster setup is appropriate for the cluster.

   📖 For information about installing the application server and configuring the SilverMaster database, see *Installing Novell exteNd*.

2. Add any deployment databases or connection pools your system needs.

3. Using the installation program, install the clustering components on one or more machines. You can use a combination of platforms (UNIX, NetWare, and Windows).

   📖 For more information, see *Installing Novell exteNd*.

4. Start the clustering components.

   📖 For more information, see “Starting the clustering components” next.

   If you installed the components as daemons or services, you can stop and restart them in this mode.

5. Using the installation program, install each of the additional servers that are part of the cluster.

   📖 For more information, see “Installing cluster servers” on page 203.

6. Start the SMC and create the cluster.

   📖 For more information, see “Creating the cluster” on page 205.

7. After creating the cluster, restart all servers and components to activate the cluster.

   📖 For more information, see “Restarting the clustered servers” on page 208.

Starting the clustering components

If the load-balancing software is not currently running as a service on your network, you need to run the programs manually on the resident machine(s). If you are using the application server’s Load Manager, execute the commands in the order shown in the procedure that follows.

➤ **To start the cluster components:**

In Windows and UNIX, the programs specified in the steps below are in the server’s \bin directory. In NetWare, enter the command from the System Console.

1. Start the Cache Manager program.

Operating system	Command
NetWare	`SilverCacheMgr`
UNIX	`/bin/SilverCacheMgr`
Windows	`\bin\SilverCacheMgr.exe`

2. Start the Dispatcher program.

Operating system	Command
NetWare	`SilverDispatcher`
UNIX	`/bin/SilverDispatcher`
Windows	`\bin\SilverDispatcher.exe`

3. Start the Load Manager program.

Operating system	Command
NetWare	`SilverLoadMgr`
UNIX	`/bin/SilverLoadmgr`
Windows	`\bin\SilverLoadMgr.exe`

You can specify the JVM to start with each of these executables. For more information, see "Specifying the JVM to use" on page 72.

Using startup parameters with the Dispatcher You can start the Dispatcher with the parameters described below:

Parameter	Description
-p *propfile*	The name and location of an alternate Dispatcher.ddl file. The Dispatcher.DDL file is created by the SMC at cluster configuration time. It specifies the default ports for RMI, HTTP, and HTTPS for the Dispatcher and is located in the server's \Resources directory. Using an alternate Dispatcher.ddl file—or editing this file outside the SMC—is not recommended.
-c *upload-certificate*	Run the Dispatcher in upload-certificate mode. For more information, see "Installing certificates in a cluster" on page 215.
-h *host*	The host name specified here is converted to an IP address via: `InetAddress.getByName(host_name);` When specified, the Dispatcher will open a server socket that is listening on this IP address only. If not specified, the socket will listen on all IP addresses of the local machine.
+cp:p *path*	Prepends specified *path* to the classpath. Don't use this debugging option without first contacting Novell exteNd Technical Support. Instead, use AGCLASSPATH to make additional Java classes available to applications. For more information, see "Setting the AGCLASSPATH variable" on page 96.
+cp:a *path*	Appends specified *path* to the classpath. This option makes additional Java classes available to applications by appending the specified path to the classpath. NOTE: Use the AGCLASSPATH environment variable to extend Java classes. For more information, see "Setting the AGCLASSPATH variable" on page 96.

Installing cluster servers

After you have installed the server that will contain the cluster's SilverMaster database, use the installation program to install additional servers. When installing additional servers, you need to point them to the first server you installed (which contains the SilverMaster database), since all servers in a cluster need to use the same SilverMaster.

This section provides some installation tips. For more detailed information, see *Installing Novell exteNd*.

Guidelines for installing cluster servers (NetWare)

One step in setting up a cluster is installing the application server on each host machine that will participate in the cluster. During these installs, you'll need to follow the guidelines below to properly fill in the Database Information screen.

Installing the first application server in the cluster In a cluster, only the first application server has a SilverMaster database. When you install that server, fill in the Database Information screen as usual. For example (installing on machine HostA):

Setting	What to specify when installing on Host A
MySQL Database Host	**HostA**
MySQL Database Port	**3306**
DB User Name	**appserver**
DB User Password	*********
Confirm User Password	*********
SilverMaster Name	**SilverMaster50**
Execute SilverMasterInit	[checked]

Installing the other application servers in the cluster The other application servers in a cluster use the SilverMaster database of the first server. When you install these other servers, fill in the Database Information screen to point to that database and uncheck the Execute SilverMasterInit setting (to prevent reinitializing the SilverMaster system tables in that database). For example (installing on machine HostB):

Setting	What to specify when installing on Host B
MySQL Database Host	**HostA**
MySQL Database Port	**3306**
DB User Name	**appserver**
DB User Password	*********
Confirm User Password	*********
SilverMaster Name	**SilverMaster50**
Execute SilverMasterInit	[unchecked]

After adding each server to the cluster, you can use the SMC to create the cluster, as described in "Creating the cluster" below.

Guidelines for installing cluster servers (UNIX)

Run the application server install on the machine you want to add to the cluster. Note that:

- When you enter the SilverMaster name and other database information, you must specify the SilverMaster used by the first server you installed for the cluster.
- Each machine in the cluster must be a mirror copy of the other. This means that:
  - You must have the identical ODBC names for the SilverMaster and any additional databases that your SilverMaster references on all servers in a cluster.
  - Any necessary DBMS client software that is required must be installed on all systems.
  - JDBC drivers must reside on all servers participating in the cluster.
- Your ability to include a UNIX server in a cluster with Windows servers may be affected by the availability of database drivers.
- You should choose not to upgrade your databases.
- To bypass rerunning SilverMasterInit, you should choose not to configure your application server.

- If you have already added this server to your cluster using the SMC, the port value you specify must match the value you entered when you added the server to the cluster.

If an error displays, you probably misentered information about the existing SilverMaster database. You'll need to respecify that information.

After adding each server to the cluster, you can use the SMC to create the cluster, as described in "Creating the cluster" below.

Guidelines for installing cluster servers (Windows)

Run the application server install on the machine you want to add to the cluster. Note that:

- When you enter the SilverMaster name and other database information, you must specify the SilverMaster used by the first server you installed for the cluster.
- Each machine in the cluster must be a copy of the other. This means that:
  - You must have the identical ODBC names for the SilverMaster and any additional databases that your SilverMaster references on all servers in a cluster.
  - Any necessary DBMS client software that is required must be installed on all systems.
  - JDBC drivers must reside on all servers participating in the cluster.
- You should choose not to upgrade your databases.
- To bypass rerunning SilverMasterInit, you should choose not to configure your application server.
- If you have already added this server to your cluster using the SMC, the port value you specify must match the value you entered when you added the server to the cluster.

If an error displays, you probably misentered information about the existing SilverMaster database. You'll need to respecify that information.

After adding each server to the cluster, you can use the SMC to create the cluster, as described in "Creating the cluster" below.

Creating the cluster

Once you have multiple servers pointing to a single SilverMaster, you can create and configure the cluster. An application server can be included in only one cluster.

You **must** use an HTTP port (not an HTTPS port) to create a cluster. If you have configured an administration port, that must be the port you use.

📖 For more information, see "Using separate ports with your firewall" on page 79.

➢ **To create a cluster:**

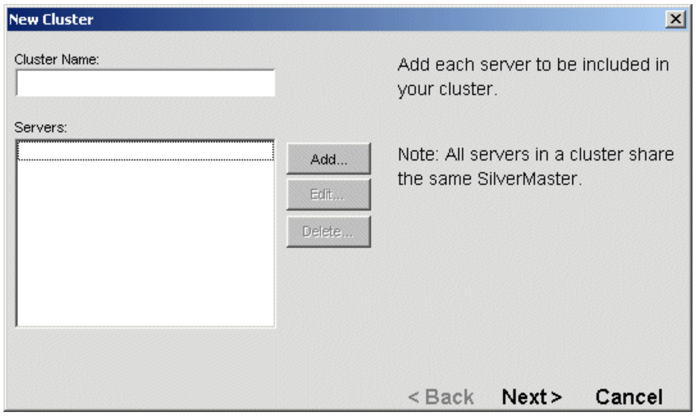
1 Make sure the Cache Manager, Dispatcher (if used), and Load Manager (if used) are running.

📖 For more information, see "Starting the clustering components" on page 202.

2 Start the SMC.

3 Click **New** (cluster) from the toolbar.

The New Cluster Wizard displays:


New Cluster

Cluster Name:

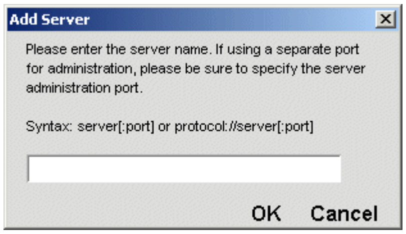
Servers:

Add each server to be included in your cluster.

Note: All servers in a cluster share the same SilverMaster.

4 Enter a cluster name, then click **Add**.

The following panel displays:


Add Server

Please enter the server name. If using a separate port for administration, please be sure to specify the server administration port.

Syntax: server[:port] or protocol://server[:port]

5 Enter an appropriately qualified name followed by a port number and then click **OK**. (The server name and the way you qualify it should match what the server is listening on. Specify the name as echoed on the server console.) For example:

```
agserver.myco.com:50001
```

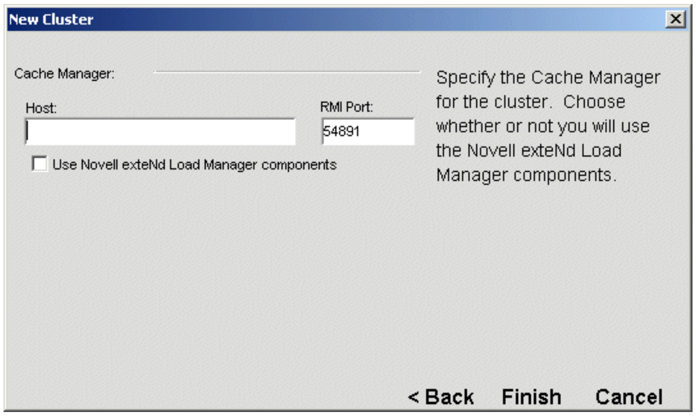
Be sure the first server you add is the one containing the SilverMaster. All subsequent servers you add must be configured to use the same SilverMaster.

If your port(s) are set to 80, you don't need to specify a port number. However, if you have defined an administration port, be sure to specify that port number. If the application server is not your primary Web server, you must change the port to a number above 5000. For instructions on changing ports, see "Specifying general server properties" on page 80.

6 Click **Add** again and enter the server name for as many servers as you intend to add to the cluster. Each server is listed in the New Cluster form as you add it to the cluster.

7 Click **Next** to set up your Cache Manager.

The following panel displays:


New Cluster

Cache Manager:

Host:

RMI Port: 54891

□ Use Novell exteNd Load Manager components

Specify the Cache Manager for the cluster. Choose whether or not you will use the Novell exteNd Load Manager components.

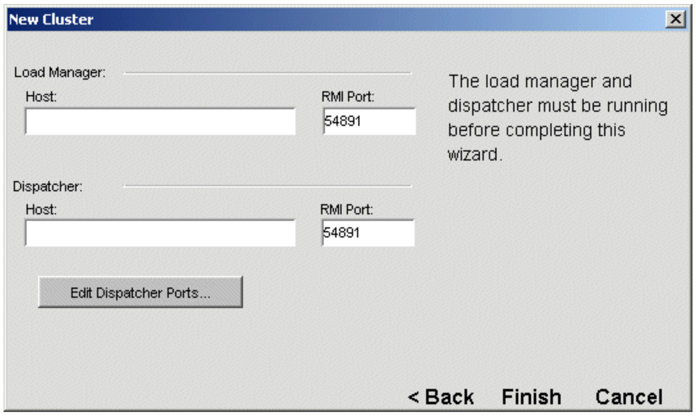
8 Enter the host name of the Cache Manager. The default RMI port number for the Cache Manager is 54891. When initially creating the cluster, you should specify the default port, but you can later change the port if necessary.

For information about changing the default ports, see "Changing the clustering components' properties" on page 214.

9 If you plan to use the application server's Load Manager, select the **Use Novell exteNd Load Manager components** check box and go to the next step.

If you do not plan to use the Load Manager at this time, click **Finish** and go to "Restarting the clustered servers" on page 208.

10 Enter the host name of the Load Manager. The default RMI port is 54891. When initially creating the cluster, you should specify the default port, but you can later change the port if necessary.



Load Manager:
Host:
RMI Port: 54891

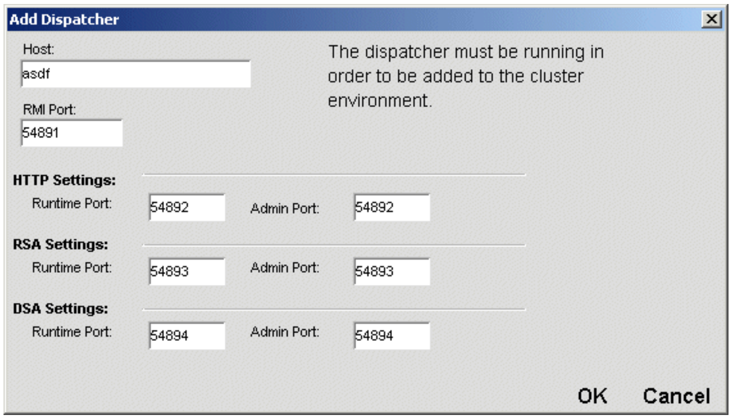
Dispatcher:
Host:
RMI Port: 54891

The load manager and dispatcher must be running before completing this wizard.

For information about changing the default ports, see "Changing the clustering components' properties" on page 214.

11 Click **Edit Dispatcher ports** to add the Dispatcher.

The following panel displays:



Host: asdf

RMI Port: 54891

The dispatcher must be running in order to be added to the cluster environment.

HTTP Settings:
Runtime Port: 54892 Admin Port: 54892

RSA Settings:
Runtime Port: 54893 Admin Port: 54893

DSA Settings:
Runtime Port: 54894 Admin Port: 54894

12 Enter the host name for the Dispatcher.

13 Specify port settings for any or all of the ports in each protocol type you want the Dispatcher to listen on.

The Dispatcher listens on all configured port types: HTTP, RSA, and DSA for any of the following unique server ports that you have already configured and enabled:

- A runtime port for users executing application objects in the cluster
- An administration port for use with SMC operations

You can disable HTTP port(s) and use HTTPS or RMI client communications. For more information, see "Turning off HTTP communications" on page 150.

For information about default port settings for each security protocol, see “Port types” on page 80.

The following table describes the port settings that appear in the panel. When initially creating the cluster, you should specify the default ports for each. You can change the ports later if necessary.

Item	Description
RMI Port	The port for Dispatcher communications with the Load Manager. Used by the Dispatcher, Cache Manager, and Load Manager. The default is 54891.
HTTP Settings: Runtime Port Admin Port	The HTTP port(s) for unencrypted communications with clients used by the Dispatcher. You can configure up to three HTTP ports. By default, the cluster runtime and administration ports use the same default port number: 54892. See “Turning off HTTP communications” on page 150.
RSA Settings: Runtime Port Admin Port	The RSA port(s) for encrypted communications (using HTTPS) used by the Dispatcher. By default, the Dispatcher uses the same default port number (54893) for all RSA runtime and administration ports in the cluster.
DSA Settings: Runtime Port Admin Port	The DSA port(s) for encrypted communications (using HTTPS) used by the Dispatcher. By default, the Dispatcher uses the same default port number (54894) for all DSA runtime and administration ports in the cluster.

For information about changing the default ports, see “Changing the clustering components’ properties” on page 214.

14 Click **OK**. You return to the New Cluster panel.

15 Click **Finish**.

Restarting the clustered servers

After creating the cluster, you must restart each server. If you are running the Load Manager, you must also restart the Load Manager and Dispatcher.

➤ To restart a server:

1 Select the server in the left panel in the SMC.

2 Click **Restart** (server).

For more information about restarting servers, see “Restarting the application server” on page 74.

➤ To restart the Load Manager and the Dispatcher:

- See “Starting the clustering components” on page 202.

Administering a server cluster

After creating a cluster, the SMC displays options specific to a clustered environment.

About properties in a clustered environment

In a clustered environment, there are three types of properties:

Property type	Description
Server local properties	Properties that are specific to an individual server and stored in the server's httpd.props file (and not the SilverMaster). These are properties that are needed by a server in order to start, so they are stored externally and are available at server start. 📖 For a listing of these properties and information on setting them, see Appendix A, "The httpd.props File".
Server stored properties	Properties that are specific to an individual server and stored in the SilverMaster database (in the AgProperties table). For a listing of these properties, select a **server** in a cluster in the SMC and look at the properties listed in the panels. The listed properties include the server stored properties as well as the server local properties that are configurable in the SMC. All server stored properties are configurable in the SMC.
Cluster shared properties	Properties shared by all servers in the cluster. These properties are stored in the SilverMaster database (in the AgProperties table). These are cluster-level properties: all servers in the cluster share the same values for the cluster shared properties. Most of the security properties are cluster shared properties. For a listing of these properties, select a **cluster** in the SMC and look at the properties listed in the panels. All cluster shared properties are configurable in the SMC.

How properties are set when a cluster is created or dissolved

When a cluster is created, any server in the cluster retains its **server local** and **server stored** properties from when it was originally configured as a standalone server. You can choose to retain these settings or change them at the server level.

📖 For more information, see "Setting server-level properties in a cluster" on page 211.

However, a server included in a cluster does **not** retain the values for properties that are defined as **cluster shared** properties. Because individual servers in the cluster may not have the same **cluster shared** property values as they did when they were standalone servers, you must reconfigure the **cluster shared properties** at the cluster level.

Accordingly, when you create a new cluster, all the **cluster shared** values are set to the default values. You can keep these settings or change them at the cluster level. Once you change a cluster-level property, the new value is applied to all the servers in the cluster. When a cluster is dissolved, all servers in the cluster become standalone servers.

📖 For more information, see "Setting cluster-level properties" next.

Setting cluster-level properties

As mentioned above, when you are working in a clustered environment, some properties exist at the cluster level and some exist at the server level. If you select the cluster at the left-hand side of the SMC, you see the cluster-level properties. Most cluster properties are the same as those set for standalone servers. The following table provides cross-references to the documentation for the cluster-level properties.

Configuration properties

The cluster-level configuration properties are grouped in several SMC panels:

Panel	Description
General	RMI/ORB and SSL for remote object properties for the cluster. 📖 For more information, see "Specifying ORB settings" on page 84.
Advanced	Performance, Cache Manager, and Load Manager properties: ◆ **Performance** Timeout on server request; session timeout. See "Setting performance parameters" on page 181. ◆ **Cache Manager** These properties exist only in clusters. See "Managing component failover" on page 212. ◆ **Load Manager** These properties exist only in clusters. See "Managing component failover" on page 212.
Managers	Cache Manager, Load Manager, and Dispatcher properties. These are the properties you specified when you created the cluster. You can edit these properties after creating a cluster. 📖 For information about these properties, see "Creating the cluster" on page 205. 📖 For information about changing these properties after you have created the cluster, see "Changing the clustering components' properties" on page 214.
Servers	Lets you add servers to and remove servers from the existing cluster and change a server's load weight.

Security properties

The cluster-level security properties are grouped in the following SMC panels:

Panel	Description
General	Specify general security settings for the cluster. 📖 For more information, see "Setting Up Security" on page 119.
Advanced	Specify client certificate levels and trusted clients list for the cluster. 📖 For more information, see "Setting Up Security" on page 119.
Permissions	Read cluster configuration, modify cluster configuration, and set permissions for the cluster. 📖 For more information, see "Setting Up Security" on page 119.
Users & Groups	Manage Silver Security and certificate users and groups for the cluster. 📖 For more information, see "Setting Up Users and Groups" on page 89.

Panel	Description
Certificates	View certificates that have been installed on the server and recognized Certificate Authorities (CAs). For more information, see “Setting Up Security” on page 119.
Security Providers	Configure external security providers, including Windows directory services, LDAP, NIS+, and certificate issuers. For more information, see “Setting Up Security” on page 119.

Monitor properties

The cluster-level monitor properties are a subset of the properties for standalone servers.

For more information about the monitor properties, see “Monitoring server activity” on page 100.

Setting server-level properties in a cluster

When a standalone server is added to a cluster, many of its server-level properties become cluster-level properties.

If you select a server in a cluster at the left-hand side of the SMC, you see the properties for a **server** in a cluster. Most properties for servers in a cluster are the same as those set for standalone servers.

Configuration properties

This section provides cross-references to the documentation for the properties of servers in clusters. The configuration properties of a server in a cluster are grouped in the following SMC panels.

Panel	Description
General	General settings for the server. For more information about the general properties, see Chapter 5, “Running the Server”.
Advanced	Debug, Performance, Cache, Transactions, Cache Manager, and Load Manager properties. ◆ **Debug** See “Low-level debugging” on page 219. ◆ **Performance** See “Setting performance parameters” on page 181. ◆ **Cache** See “Managing the server content cache” on page 187. ◆ **Transactions** See “Managing J2EE transactions” on page 99. ◆ **Cache Manager** These properties exist only in clusters. See “Managing component failover” on page 212. ◆ **Load Manager** These properties exist only in clusters. See “Managing component failover” on page 212.
Pools	Connector and JDBC connection pools. For more information, see “Configuring connection pools” on page 51.
Connections	Client connection properties. For more information, see “Managing client connections” on page 182.

Panel	Description
Databases	Information about the deployment databases in the cluster (databases known to the cluster's SilverMaster). You can modify the minimum and maximum number of database connections for each server in the cluster. For more information, see "Configuring deployment databases" on page 46.

Security properties

The accelerator settings for a server in a cluster are administered at the server level.

For more information about accelerator settings, see "Using Cryptographic Hardware Integration" on page 161.

Monitor properties

The monitor properties in a cluster are the same as the properties for standalone servers.

For more information about the monitor properties, see "Monitoring server activity" on page 100.

Specifying a server's relative load weight

You can specify the relative processing weight for each server in a cluster. The Load Manager uses that information to generate a distribution map that determines the processing load for each server at runtime.

For a description of how weighting works, see "Distribution mapping" on page 198.

➤ **To specify a server's relative load weight:**

1. Select the cluster in the SMC.
2. Select the **Configuration** icon from the toolbar, then select **Servers**.
3. Select a server from the list and specify an integer in the **Server Load Weight** field.
4. Click **Update**.
5. Select the other servers and specify appropriate relative values.
6. To activate the new server weight settings, click the **Restart** (server) button for the servers.

Managing component failover

The SMC provides access to properties that control how the Cache Manager and Load Manager respond in the event of a system failure. Normally you do not need to edit these properties.

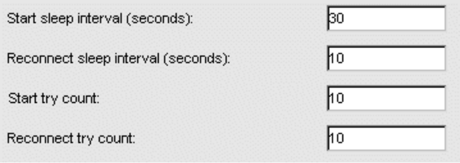
The Cache Manager and Load Manager properties exist at both the cluster and server levels. You can set the properties at the cluster level, which sets the values for each of the servers in the cluster. You can then override the values for any of the servers in the cluster. But note that if you later change any of the properties at the cluster level, it will override settings you made at the server level.

Cache Manager properties

The Cache Manager properties determine how the Cache Manager will respond when its connection with a server fails.

➢ **To set Cache Manager properties:**

1 Start the SMC.
2 Select the cluster to set properties at the cluster level, or select a server within a cluster to set properties at the server level.
3 Select the **Configuration** icon from the toolbar.
4 Select **Advanced**.
5 Select the **Cache Manager** tab:



Start sleep interval (seconds): 30
Reconnect sleep interval (seconds): 10
Start try count: 10
Reconnect try count: 10

6 Reset any of the properties:

Property	Description
Start sleep interval	The number of seconds the Cache Manager will delay before starting a series of attempts to reconnect with the server.
Reconnect sleep interval	The number of seconds before a new series of reconnection attempts.
Start try count	The number of times to start a series of reconnection attempts before generating an error.
Reconnect try count	The number of times to attempt to reconnect in a series.

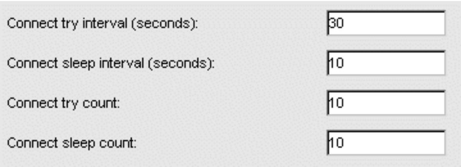
7 Click **Update**
8 To activate the new properties, click the **Restart** (server) button.

Load Manager properties

The Load Manager properties determine how the Load Manager will respond if its communication with a server fails.

➢ **To set Load Manager properties:**

1 Start the SMC.
2 Select the cluster to set properties at the cluster level, or select a server within a cluster to set properties at the server level.
3 Select the **Configuration** icon from the toolbar.
4 Select **Advanced**.
5 Select the **Load Manager** tab:



Connect try interval (seconds): 30
Connect sleep interval (seconds): 10
Connect try count: 10
Connect sleep count: 10

6 Reset any of the properties:

Property	Description
Connect try interval	The number of seconds to delay after each retry series
Connect sleep interval	The number of seconds to delay after each connection retry in a series
Connect try count	The number of times the Load Manager should begin a series of connection attempts with the server
Connect sleep count	The number of connection retries in a series

7 Click **Update.**

8 To activate the new properties, click the **Restart** (server) button.

Dissolving a cluster

You can dissolve a cluster, which does the following:

- Deactivates the cluster
- Deletes the cluster

➢ **To dissolve a cluster:**

1 Click **Dissolve** (cluster) in the SMC toolbar.

2 Click **OK**.

Changing the clustering components' properties

You can change which hosts are running the clustering components: Load Manager, Dispatcher, and Cache Manager. You can also change which ports the clustering components will use.

Changing hosts

After creating a cluster, you may decide to run the clustering components on different hosts.

➢ **To change the hosts:**

1 In the SMC, select the cluster.

2 Select the **Configuration** icon from the toolbar.

3 Select **Managers**.

4 Update the hosts as needed for the clustering components.

5 Click **Update**.

6 Stop and restart each server and each clustering component in the cluster.

Changing ports

By default, all clustering components use port 54891 for their RMI port. In addition, the Dispatcher uses ports 54892, 54893, and 54894 for its HTTP, RSA, and DSA ports respectively. Normally, you don't change these port values.

But if you need to, you can change ports for the Cache Manager, Load Manager, and Dispatcher after you create the cluster.

NOTE: All clustering components must use the same RMI port.

➢ **To change Cache Manager, Load Manager, and Dispatcher ports:**

1. In the SMC, select the cluster.
2. Select the **Configuration** icon from the toolbar.
3. Select **Managers**.
4. Update the port specifications.
5. Click **Update**.
6. Stop and restart each server and each clustering component in the cluster.

   If you changed the port for the Cache Manager, you must start it with the following command line:

   ```
   SilverCacheMgr -p portNumber
   ```

Changing the administration port of a server in a cluster If the server is in a cluster, and you need to change the administration port, you must:

1. Dissolve the cluster.
2. Change the administration port.
3. Rebuild the cluster.

Installing certificates in a cluster

For a cluster to listen and serve on the HTTPS port, you must install a server certificate in each server in the cluster. If you are using the application server's software Dispatcher (**SilverDispatcher**), you also need to install a certificate for it.

📖 For more information about certificates and HTTPS/SSL, see "Using certificates" on page 136.

About the server certificates

If you are using the application server's software Dispatcher for the cluster, each application server must have a server certificate with the DNS name matching the **server's** host name.

If you are using a third-party hardware dispatcher (such as Cisco LocalDirector) that does URL masking—that is, masks all URLs such that the browser hitting any of the servers in the cluster sees the dispatcher's host name—then every application server must have a server certificate with the DNS name matching the **dispatcher's** host name.

For example, say you have a cluster with:

- A Dispatcher with the host name **www.myhost.com**
- Three servers with host names **server1**, **server2**, and **server3**

If you are using the application server's Dispatcher, you need to create four server certificates: one for the Dispatcher's machine (with the DNS name **www.myhost.com**) and one for each of the servers (with DNS names **server1.myhost.com**, **server2.myhost.com**, and **server3.myhost.com**).

If you are using a third-party URL-masking dispatcher, you need to create only one server certificate (with the DNS name **www.myhost.com**) and upload it to each of the servers.

How to do it

The following procedures describe what to do.

➢ To generate server certificates:

- Use the SMC to generate an RSA or DSA certificate.
  - If you are using the application server's software Dispatcher, generate a separate certificate for each server used in the cluster and for the Dispatcher.
  - If you are using a third-party dispatcher, generate one certificate with the dispatcher's DNS name, as described above.

  📖 For more information, see "About certificates" on page 136.

➢ To install certificates if you are using the software Dispatcher:

1. Use the SMC to install the certificates on the server.

   📖 For more information, see "Creating and installing server certificates using the SMC" on page 138.

2. After the certificate has been installed, restart the server.

   The server is now listening on its HTTPS port (default: 443 for RSA and DSA certificates).

3. Repeat Step 1 and Step 2 for each server in the cluster.
4. To install the certificate on the Dispatcher, start the Dispatcher using the -c startup option. This puts the Dispatcher in the mode in which it can upload a certificate.

   📖 For more information, see "Starting the clustering components" on page 202.

5. Invoke **AgDigitalIDStep2** to install the certificate on the machine containing the Dispatcher.

   📖 For more information, see "Using AgDigitalIDStep2" on page 148.

6. When prompted, specify the machine containing the application server's Dispatcher and specify the HTTP port that the Dispatcher is listening on (default: 54892).
7. After the certificate has been installed, stop the Dispatcher and then restart it as normal (without the -c startup option).

   The Dispatcher is now listening on its HTTPS port (default: 54893 for RSA certificates and 54894 for DSA certificates).

➢ To install certificates if you are using a third-party dispatcher:

1. Invoke **AgDigitalIDStep2** to install on a server the certificate that references the dispatcher's DNS name.

   📖 For more information, see "Using AgDigitalIDStep2" on page 148.

2. When prompted, specify the server and the HTTP port that the server is listening on (default: **83** in NetWare, **80** in NT, **8080** in UNIX).
3. After installing the certificate, stop the server.
4. Add the following line to the server's **httpd.props** file (in the server's **\Resources** directory):

   ```
   http-server.com.sssw.srv.https.cert.hostname=DispatcherName
   ```

   where *DispatcherName* is the DNS name of the dispatcher.

   📖 For more information about httpd.props, see Appendix A, "The httpd.props File".

5. Restart the server.

   The server is now listening on its HTTPS port (default: 443 for RSA and DSA certificates).

6. Repeat the preceding steps for each server in the cluster.

13 Using the Server Administration API

The Server Administration API allows you to programmatically configure and manage a Novell exteNd Application Server. Applications written with this API can be deployed as Java applications running on the server or as Java clients, such as SilverJ2EEClient. The application server's SMC (which uses a Java client) is built with this API. You can use the Server Administration API to programmatically handle these tasks:

- Load balancing configuration
- Set security on objects
- Manage deployed J2EE archives
- Manage server sessions
- Manage certificates
- Manage threads
- Manage statistics

This chapter has the following sections:

- Status of the Server Administration API
- For information about the Server Administration API

Status of the Server Administration API

The Server Administration API will likely be replaced in the future by the **J2EE Management API**.

The J2EE Management API will result from JSR-77 and be part of J2EE 1.4. It will provide a standard model for managing J2EE environments.

📖 For more information about the J2EE Management API, see http://jcp.org/jsr/detail/77.jsp.

For information about the Server Administration API

To obtain the Server Administration API documentation, contact Novell exteNd Technical Support.

14 Troubleshooting

This chapter describes some techniques and procedures that you can use for troubleshooting the Novell exteNd Application Server. This chapter contains the following sections:

Using error logging

Consider turning on error logging at all times when running the server. Error logging is a lightweight process that prints detailed information about error messages either to the AgErrorLog table in the SilverMaster or to a file you designate. You can activate logging using the SMC. For more information, see "Using server logging" on page 82.

When logging to the SilverMaster, you can view the log in the SMC by selecting the **Monitor** icon from the toolbar and then selecting **Logs**.

📖 For more information, see "Displaying logs" on page 103.

Low-level debugging

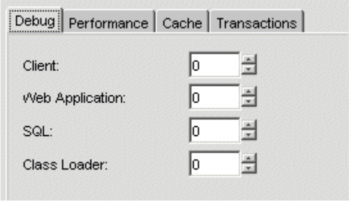
The Debug options in the SMC enable the printing of server debug messages to the server console. Options include debugging client requests, Web applications, and SQL statements. You should activate debugging options only for application debugging purposes, as this activity can significantly inhibit server performance.

NOTE: If you are running the server as a service in Windows, the output is printed to the error log instead of the console window.

➢ **To print debugging messages:**

1. Start the SMC.
2. Select the **Configuration** icon from the toolbar.
3. Select **Advanced**.

4 Select the **Debug** tab:



Client: 0

Web Application: 0

SQL: 0

Class Loader: 0

5 Change the value for the type of activity you are debugging.

The number you enter indicates the level of detail you want displayed. The value 0 means that messages are not printed. You have the following debugging options:

Field	Description
Client	If this parameter is set to **1**, the server logs information for each client (such as http GETs, PUTs, and POSTs). If this parameter is set to **2 or greater**, the server logs the **complete** request and reply message to the console window. TIP: This option is useful for problems with HTTP, servlets, and other client-related issues.
Web Application	If this parameter is set to **1 or higher**, the server logs information about the execution of each running Web application as follows: ◆ If set to **1 or 2**, provides minimal or maximal information about methods called in the public API. ◆ If set to **3 or 4**, provides minimal or maximal information about all methods called, even methods not in the public API. ◆ If set to **5**, echoes all output to ServletOutputStreams so that you can see what is being output from the business object. ◆ If set to **6**, includes stack traces of various calls so that you can see where calls are being made.
SQL	If this parameter is set to **1**, the server logs each SQL statement executed against the database for client data. If this parameter is set to **2 or greater**, the server logs additional information that Technical Support can use to track down server problems. TIP: This option is useful for debugging database-related problems.
Class Loader	If this parameter is set to **1 or higher**, the server logs information about the J2EE ClassLoader (used for each J2EE application) as follows: 1—Shows only the list of repositories (places that the ClassLoader looks to find things) when they are added. Use this to find out where the ClassLoader will be looking for things. 2—Shows arguments to the basic API calls like findClass(), loadClass(), getResource(), and so on. Use this to find out when basic API calls are made. 3—Shows more internal information. Use this setting when tracing the search for a class or resource through the various search locations. 4—Shows stack traces in key areas. Use this when you need to know who is loading a particular class.

Setting JDBC tracing

If you have persistent problems with a database connection, use JDBC tracing.

➢ To set JDBC tracing:

1. If necessary, create a log file for storing trace data.
2. Shut down the server (see "Shutting down the application server" on page 74).
3. Open the **httpd.props** file (located in server's **\Resources** directory).
4. Add an **http-server.Jdbc.DriverManager.LogFile** entry to the props file and point it to the log file. For example, if the log file is d:\test\jdbc.log, create this line in your httpd.props file:

   ```
   http-server.Jdbc.DriverManager.LogFile=d:\\test\\jdbc.log
   ```

5. Restart the server.

   NOTE: Use JDBC tracing for troubleshooting only, as it will slow down the server and use considerable disk space.

Using the server's command shell

You can type commands into the server's console window (the window in which the server was started) to obtain diagnostic information about the state of the server. For example, you can get information on memory usage, threads, sessions, and server system properties. You can also enable tracing for different subsystems.

➢ To learn more after the server has started:

- In the console window, enter:

  ```
  help
  ```

  All available commands are listed, and the command line is marked by a ! character (you can change the character using the **prompt** command).

➢ To display help on a command:

- Enter:

  ```
  help command
  ```

➢ To disable the command shell:

1. Enter this line in the httpd.props file:

   ```
   http-server.com.sssw.srv.commandshell=false
   ```

2. Restart the server.

Using the Watcher

The Watcher tool helps you understand the state of the server in cases when the server becomes unresponsive. When it is activated, the Watcher logs the state of the server once a minute.

TIP: You might find the Watcher valuable when faced with problems that are hard to debug.

➢ To use the Watcher:

1. Add the following property to the **httpd.props** file:

   ```
   http-server.com.sssw.srv.httpdwatcher
   ```

2. Set the value of the property to the pathname of a watcher configuration file. For example:

```
http-server.com.sssw.srv.httpdwatcher=c:\\temp\\watchconfig.txt
```

TIP: Remember to escape backslashes in the **httpd.props** file.

What happens

If this property is set when the server is started, the server will create a *watcher thread*. The watcher thread sleeps, waking up once a minute to check for the existence of the watcher configuration file supplied as the value of the **httpdwatcher** property.

If the watcher configuration file does not exist As long as the configuration file doesn't exist, the Watcher does nothing and just goes back to sleep. Under these circumstances, the Watcher has minimal impact on server performance. And even if the server hangs, in most cases the Watcher will not hang.

If the watcher configuration file exists When the Watcher discovers that the watcher configuration file exists, it reads the configuration file and uses it to control its further actions.

About the watcher configuration file The watcher configuration file is an ASCII text file that must include:

- A **flags value** that tells the Watcher what information to print

  The flags value is a bit-coded integer in which the bits are defined as follows:

  - Bit 0 (== 0x1) dumps threads info
  - Bit 1 (== 0x2) dumps session info
  - Bit 2 (== 0x4) dumps database connection info
  - Bit 3 (==0x8) not used
  - Bit 4 (==0x16) dumps thread event log
- An (optional) **output file name** for the Watcher output (if not supplied, the output goes to the server console)

A **typical watcher configuration file** looks like this:

```
7
c:\temp\watchout.txt
```

This file tells the Watcher to dump information once every minute about threads, sessions, and database connections to the specified output file.

Common problems starting the server

This section describes some reasons the application server might fail to start, and how you can address the problem. For more information on troubleshooting server problems, see "Using SilverMasterInit to recreate or refresh SilverMaster" on page 229.

NOTE: Server failure is often related to the specific database you are using. For database-specific information, see the *Database Configuration Guide*.

System resource problems

There are two causes of server failure that are due to inadequate system resources:

Cause	Description	What to do
Insufficient disk space	The operating system needs more space to write files to disk	Create more disk space by eliminating or moving files, then try starting the server again
Insufficient memory	This might be a temporary problem or it might indicate insufficient resources	Shut down other programs, expand your swap file, or add more memory to the server machine

Database not synchronized

If you see errors related to database consistency, go to the SMC and execute the **Synchronize database schema** option, then restart the server.

📖 For more information, see "Synchronizing the database schema" on page 51.

Using SilverMonitor

SilverMonitor is a background process running on the server that monitors server status and attempts to restart the server if it terminates abnormally. By default, this process is activated when the server is started. SilverMonitor starts with default parameters, which you can modify. You can also run the server without SilverMonitor.

There are two ways you can modify the parameters:

- Specify parameter(s) as server startup options on the DOS command line. See "Using startup options" on page 70.
- Specify machine default values by editing the registry (Windows only).

Order of precedence

The order of precedence is as follows:

- Any values in the registry override the default values.
- Any values manually entered as startup options override registry values.

Summary of parameters

The following is a summary of SilverMonitor parameters:

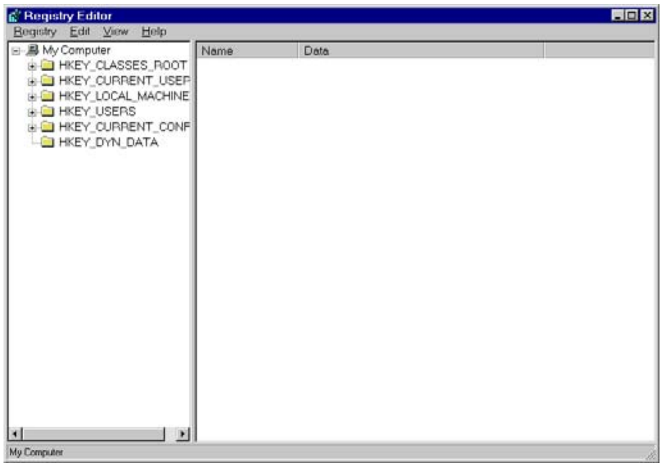
Startup option	Registry option (NT)	Description
Default	/X	Command in the registry to start SilverMonitor.
-retry *number*	/C_number	Number of restart tries. The default is 3.
-minspan *number*	/M_*minutes*	Minute span for restart tries. The default is 10.
—	/D	Debug information about the SilverMonitor process.
-nomonitor	—	Run the server without SilverMonitor.

➢ **To modify SilverMonitor parameters in the NT Registry:**

1 From the Start menu, choose **Run**.

2 Type **regedit**.

The Registry Editor displays:



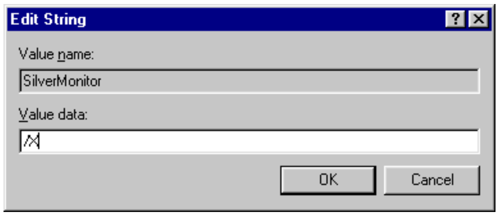
3 Navigate the tree as follows:

HKEY_LOCAL_MACHINE>SOFTWARE>Novell>exteNd>
AppServer>*version number*

4 Double-click the **SilverMonitor** process.

5 When the following dialog displays, type one or more of the options described above. Separate each option with a space.

The following shows the option for starting SilverMonitor:



Value name: SilverMonitor

Value data: /X

6 Click **OK**.

What happens

SilverMonitor writes to the NT EventLog when it restarts the server. It also writes to a SilverMonitor.log file in the directory where it is run (usually the server's **\bin** directory). This log file gets an entry every time the monitor starts.

When you restart the SilverMonitor, the log file is emptied and restarted.

Using the SilverMasterInit program

The application server relies on the SilverMaster database for overall system management. **SilverMasterInit** is a command-line program that performs several types of processes on the SilverMaster database. The SilverMasterInit executable is located in the server's **\bin** directory. SilverMasterInit can:

- Recreate or refresh tables and properties used by the SilverMaster database
- Generate logs

- Display debug information
- Regain access to locked resources

This section contains the following topics:

- Command-line options
- Using SilverMasterInit to recreate or refresh SilverMaster
- Regaining access to SilverMaster

Command-line options

The table below describes how and when to run each of the SilverMasterInit command-line options. To see a list of options, type the following at the command prompt:

```
SilverMasterInit -?
```

Administration accounts There are two administration accounts (**database** and **server**). Both accounts are defined during installation. The server administration account restricts who can log in and administer the application server. You define the server administration account using SilverMasterInit. After a default installation, the server administrator user account is part of the predefined Administrators group and has the Locksmith privilege.

The application server uses the database administration account when connecting to the SilverMaster database. The only time you need to specify the SilverMaster database account is when you are running SilverMasterInit.

Entering options You must enter the database user account name and password for all command-line options. You also need to specify Full or Refresh mode for all SilverMasterInit options—**except** those noted in the following table. When you specify a Full mode database initialization, three options (-A, -n, and -W) require that you also define the server administration account name and password on the command line.

You specify parameter(s) as SilverMasterInit startup options on the command line.

SilverMasterInit startup option	Description	Use
-?	Displays usage for SilverMasterInit.	Use to check option usage.
+cp:a *path*	Appends specified *path* to the class path.	This option makes additional Java classes available to applications by appending the specified path to the class path. **NOTE:** Use the AGCLASSPATH environment variable to extend Java classes. Example: `SilverMasterInit [-f or -r] +cp:a path -U dbusername -P dbpassword`
+cp:p *path*	Prepends specified *path* to the class path.	Don't use this debugging option without first contacting Novell exteNd Technical Support. Instead, use AGCLASSPATH to make additional Java classes available to applications. See "Setting the AGCLASSPATH variable" on page 96.

SilverMasterInit startup option	**Description**	**Use**
-A *adminname*	Specifies the server administrator user name used to log in to and administer the application server.	This option lets you define a server administrative account name (and password) when you are creating a new SilverMaster database catalog. The server user account you specify will be part of the Administrators group and have full Locksmith privilege. Use this account to administer the server. See "About your administrator account" on page 90. When running a Full mode database initialization you must specify the server administration account name and password. Example: `SilverMasterInit -f -U` *`dbusername`* `-P` *`dbpassword`* `-A` *`adminusername`* `-W` *`adminpassword`*
-a	Causes the application server to require users to authenticate themselves.	Set this parameter if you accidentally restrict Read access to your SilverMaster database, which includes the login resource. This option also provides a quick way to set authentication without running the SMC. See "Using server authentication to access the login resource" on page 231. You don't need to specify Refresh or Full mode when running the **-a** option. Example: `SilverMasterInit -a -U` *`dbusername`* `-P` *`dbpassword`*
-b	Displays boot environment settings.	Run to see the initial SilverMaster environment properties used by Full mode or Refresh mode. Example: `SilverMasterInit [-`*`f`*` or -`*`r`*`] -b -U` *`dbusername`* `-P` *`dbpassword`*
-c	Checks that BLOBs are inserted into the database correctly.	Run to verify that these objects are properly stored. Example: `SilverMasterInit [-`*`f`*` or -`*`r`*`] -c -U` *`dbusername`* `-P` *`dbpassword`*

SilverMasterInit startup option	Description	Use

-C *properties file*	Reads the nameServicePort or IIOP SSL ports properties from the specified *properties file*.	The *properties file* has the same syntax as the httpd.props file, except that you do not specify http-server in front of the property. Instead you specify
		`<hostname>.<portnumber>`
		The *hostname* and *portnumber* are separated by a dot (not a colon).
		The *portnumber* is the administration port and is only required when it is some value other than 80.
		To set the nameServicePort, set the following property:
		`<hostname>.<portnumber>.com.sssw.srv.`
		`nameServicePort=<nameServicePort>`
		For example:
		`tundra.8080.com.sssw.srv.nameServicePort=55597`
		To set the IIOP SSL port range, use these properties:
		`<hostname>.<portnumber>.com.sssw.srv.`
		`port_iiop_ssl_min`
		`<hostname>.<portnumber>.com.sssw.srv.`
		`port_iiop_ssl_max`
		The port_iiop_ssl_min property specifies the lower bounds for IIOP SSL. If you do not specify a range, the ORB picks the first available port. Use -1 if you don’t need to specify a range. The port_iiop_ssl_max property specifies the upper bounds for IIOP SSL communications. Use -1 for no upper bound.
		Run SilverMasterInit with the -r option to update an existing SilverMaster.
		Example:
		`SilverMasterInit -C port.props -U smbb -P password -A`
		`administrator -W admin -r`

-D *database*	Deletes all Ag tables from the specified SilverMaster database.	Deletes all existing application server system tables (including users, groups, and licensing data) from the specified SilverMaster database. Use to remove a deployment database from the server. Unlike Full mode, this option deletes application server system data but does not replace it with initial properties. Example: `SilverMasterInit -U dbusername -P dbpassword -D Agdb`

-e *error log file*	Writes errors to the specified file, which is created as necessary.	If no errors are found, a log file is not created.
		If you don’t specify a path, the error log file is stored in the directory from which you ran SilverMasterInit. The default file name is **sminit.log**.
		Example:
		`SilverMasterInit [-f or -r] -U dbusername -P dbpassword -e`
		`c:\ServerLogs\sminit.log`

-f	Runs in Full mode to create a new SilverMaster database.	Creates new SilverMaster system data and resources. This option deletes existing users, groups, and licensing data.
		NOTE: By default, the server will be restricted when you run Full mode. To install the server unrestricted (for a development environment), run SilverMasterInit in Full mode with **-n**.
		When running a Full mode database initialization, you must specify the server administration account name and password.
		Example:
		`SilverMasterInit -f -U dbusername -P dbpassword -A`
		`adminusername -W adminpassword`

SilverMasterInit startup option	**Description**	**Use**
-L *jdbc log file*	Writes JDBC debugging information to the specified log file.	If you don't specify a log file name, this option is ignored. If you don't specify a path, the JDBC log file is stored in the server's **\bin** directory. Example: `SilverMasterInit [-f or -r] -L c:\ServerLogs\logs\jdbclogfile.log`
-l *locksmith account*	Specifies a user or group account to grant Locksmith privilege to.	Use this option if you accidentally delete all accounts that have Locksmith privilege. See "Regaining access to SilverMaster" on page 231. You don't need to specify Refresh or Full mode when running the **-l** option. Example: `SilverMasterInit -l -U dbusername -P dbpassword`
-n	Unrestricts access to the application server.	Use when you do **not** want to lock down access to system data nor require user authentication. This option means any user can perform administrative operations and browse directory listings until you lock down access by setting permissions. See "Default server and object security" on page 172. You need to specify Full mode and the server administration user name and password when running the **-n** option. Example: `SilverMasterInit -n -f -U dbusername -P dbpassword -A adminusername -W adminpassword`
-O *table space*	Creates all Ag tables in the specified Oracle table space for SilverMaster.	Use this option when creating a SilverMaster database to use with Oracle. More space (than the default) must be allocated for SilverMaster table objects because of the way an Oracle database stores data. Example: `SilverMasterInit [-f or -r] -U dbusername -P dbpassword -O tablespacename`
-P *dbpassword*	Specifies the database password used by the application server to access SilverMaster.	The database administration password and associated user account are stored encrypted in the registry during server installation. The server will use the specified account name and password at startup to access the SilverMaster database. Example: `SilverMasterInit [-f or -r] -U dbusername -P dbpassword`
-p *properties file*	Reads startup properties from the specified file. Defaults to **httpd.props** in the server's **\Resources** directory.	Use to specify a SilverMaster startup property file name and location other than the default. After you set the property file option with SilverMasterInit, you need to start the application server from the command line with the **-p** option to use the new property file. Example: `SilverMasterInit [-f or -r] -p c:\Program Files\Novell\exteNdn\AppServer\Resources\httpd.newprops -U dbusername -P dbpassword`
-r	Runs in Refresh mode to update SilverMaster resources.	This process skips some of the database installation steps used by Full mode. Use this option to refresh SilverMaster system data and resources when you don't want to delete existing users, groups, and licensing data. Example: `SilverMasterInit -r -U dbusername -P dbpassword`

SilverMasterInit startup option	Description	Use
-W *adminpassword*	Specifies the server administrator account password used to log in to and administer the application server.	Use the server administrator user and account password to administer the server. When running a Full mode database initialization, you must specify the server administration account name and password. Example: `SilverMasterInit -f -U dbusername -P dbpassword -A adminusername -W adminpassword`
-U *dbusername*	Specifies the application server's SilverMaster database user account.	The database administration user account and associated password are stored encrypted in the Registry. The server will use the specified account name and password at startup to access the SilverMaster database. Example: `SilverMasterInit [-f or -r] -U dbusername -P dbpassword`
-v	Generates verbose output as SilverMasterInit runs.	If the process fails, run this option to identify where the failure occurred. Example: `SilverMasterInit [-f or -r] -v -U dbusername -P dbpassword`
-x	Displays SilverMaster initialization properties and then exits without starting SilverMasterInit.	Run to view local server startup properties. This option does not change or refresh properties. Use this debugging option to check for misdirected initialization settings. Example: `SilverMasterInit -x`

Using SilverMasterInit to recreate or refresh SilverMaster

The SilverMaster database, which is created during installation, can also be recreated or updated using SilverMasterInit. The SilverMaster database keeps track of all deployment databases used by the application server and also holds the application server's system tables, including those containing group, user, and licensing information. There is one SilverMaster catalog for each server or cluster.

For more information about the SilverMaster database, see "Configuring deployment databases" on page 46.

If your SilverMaster database is damaged, you can run SilverMasterInit. If you cannot start the application server, try one of the following procedures if nothing else has worked:

- Refreshing the SilverMaster database with SilverMasterInit
- Creating a new SilverMaster database with SilverMasterInit

CAUTION: *Connection problems may be due to a corrupt driver connection, damaged deployment database, or network problems. If you have questions about what is causing your server problem, call Novell exteNd Technical Support before running SilverMasterInit. Running SilverMasterInit in Full mode will delete the contents of all your existing system tables and replace them with initialized data. Do not run in Full mode if you want to preserve existing system tables, including those that contain deployed J2EE archives, group, and user data.*

Refreshing the SilverMaster database with SilverMasterInit

You can run SilverMasterInit in Refresh mode to upgrade or access SilverMaster properties. The refresh process skips some of the database installation steps used by Full mode. Run SilverMasterInit in Refresh mode to refresh SilverMaster system data and resources without deleting existing users, groups, and deployed J2EE archives.

NOTE: As part of the application server installation process, SilverMasterInit upgrades resources. You typically upgrade the application server by running the installation program.

➢ **To run SilverMasterInit in Refresh mode:**

1. Shut down the application server.
2. From server's **\bin** directory, enter:

   ```
   SilverMasterInit -r options
   ```

   You see the message **Creating Resources will take a few minutes; please wait**.
3. When SilverMasterInit completes without errors, restart the application server.

Creating a new SilverMaster database with SilverMasterInit

SilverMasterInit can often fix problems caused by someone removing or renaming a file or table that SilverMaster relies on. If you cannot start the application server or connect to the SilverMaster database, you may need to run SilverMasterInit.

While SilverMasterInit can reset corrupted SilverMaster properties, this program **cannot** repair a corrupted Registry key, configuration files, sample databases, or files associated with databases. To address these types of problems, run the installation program.

To avoid deleting all your database tables, try running SilverMasterInit in Refresh mode (before running it in Full mode) to see if that resolves the server problem.

If you run SilverMasterInit in Full mode to regenerate new SilverMaster properties, you will have to recreate any Silver Security users and groups, re-add any deployment databases, and redeploy any J2EE archives deployed to SilverMaster.

➢ **To run SilverMasterInit in Full mode:**

1. Shut down the application server.
2. From the server's **\bin** directory, enter:

   ```
   SilverMasterInit -f options
   ```

   You see the message **Creating Resources will take a few minutes; please wait**.
3. Record any errors from this command.
4. Start the application server.
5. Add any deployment databases again.
6. Recreate any users and groups.
7. Redeploy any J2EE archives that were deployed to the SilverMaster.

NOTE: If you do **not** want to lock down access to data and require user authentication, you can run the **-n** option in Full mode.

Regaining access to SilverMaster

You can use SilverMasterInit to regain access to locked resources. The SilverMaster database is where the application server stores all system resources and links to other databases. By default, any user with the Locksmith privilege has Read access permission to the SilverMaster. If all users are accidentally denied access to the SilverMaster database, no one will be able to access the application server through the SMC.

See the following sections if you suspect that Read access to SilverMaster has been restricted:

- Using the Locksmith option to access locked resources
- Using server authentication to access the login resource

Using the Locksmith option to access locked resources

By default, the administrator and any other user with Locksmith privilege can get and set data access permissions for any resource in any database, read all SilverMaster resources, and grant Locksmith privilege to users and groups. The only user able to grant the Locksmith privilege is someone who is already a Locksmith. If all accounts with the Locksmith privilege get deleted, use the SilverMasterInit Locksmith option to grant this privilege to a user to regain access to resources.

NOTE: By default, after a new installation or after you run SilverMasterInit in Full mode, an administrator account is automatically created that has Locksmith privilege.

Once a user with the Locksmith privilege can access SilverMaster, that user can unlock resources and reset access privileges.

For more information, see “Using the Locksmith privilege” on page 94.

➢ To reset Locksmith privilege:

1 Shut down the application server.

2 From server’s **\bin** directory, enter:

```
SilverMasterInit -l -U dbusername -P dbpassword
```

The Locksmith can now use the SMC to unlock resources.

Using server authentication to access the login resource

You can set server authentication from either the SMC or the SilverMasterInit command line. You should set server authentication if you accidentally restrict Read access to your SilverMaster database. If users cannot access SilverMaster, run the SilverMasterInit server authentication option to allow users to authenticate themselves when they initially connect to the server. When a user logs in to the application server from the SMC, a request for the login resource is issued. Users cannot access the login dialog if their access to SilverMaster is restricted, because they have no Read access to the database. This is because the SilverMaster has the *login* resource.

You do not need to specify Full or Refresh mode when you run the server authentication option. When you restart the server after setting server authentication, your first attempt to access the server will bring up the credentials dialog and you can log in.

➢ To set server authentication:

1 Shut down the application server.

2 From the server’s **\bin** directory, enter:

```
SilverMasterInit -a -U dbusername -P dbpassword
```

3 Restart the application server.

Users will now be prompted to log in.

Handling a stack overflow

In some obscure situations it is possible to exceed the limits of the stack, in which case the Java Virtual Machine (JVM) throws a java.lang.StackOverflowError.

About stacks

On a Windows system in the Java environment, there are at least two program stacks (and possibly more, depending on the JVM implementation), any one of which can overflow and cause a StackOverflowError to be thrown:

- There is always a hardware stack, which is used by native code in the JVM itself and by native code that is compiled from Java bytecodes by the Just In Time (JIT) compiler.
- There is always a Java bytecode stack, which is used for temporary storage of method call arguments and local variables for Java methods. This is a **soft** stack that is created and managed by the JVM.

Each thread created in the JVM has its own hardware and Java stacks.

What to do if you get a stack overflow

It is possible to alter the size of each of the stacks if it is determined that the default stack size is too small. However, the most common cause of stack overflow errors is a programming error where a method is called recursively a number of times. If this is the case, increasing the size of the stack will not fix the stack overflow problem. Before attempting to increase the stack size, verify that the code does not contain any errors of this nature. Assuming the stack overflow is not caused by an infinite recursion error, it should be possible to fix the stack overflow by increasing the stack size. To determine which stack overflowed is largely a matter of trial and error.

The size of the hardware stack is determined by the operating system using a value stored in the header of the executable. The executable (SilverServer.exe) specifies a default stack size of 256K.

Changing the stack size

In order to change the stack size, you must modify the executable header using Microsoft's EDITBIN utility. For example, to change the default stack size for SilverServer.exe to 512K, use the following command line:

```
EDITBIN /STACK:0x80000 SilverServer.exe
```

TIP: Of course, you should make a backup before modifying any executable.

Changing the Java stack size

If increasing the size of the hardware stack doesn't work, it is possible that the Java stack is the problem. In the JDK documentation, there are two command-line options affecting the stack size:

- -ss sets the maximum native stack size
- -oss sets the maximum Java stack size

The defaults were 128K and 400K respectively. Although these options are no longer documented in JDK 1.2 (Java 2), they appear to have been carried forward as nonstandard (-X) switches. To set these options for an application server executable, use +X instead of -X (the application server executables interpret + options as options to be passed to the JVM).

📖 For more information about application server startup options, see "Using startup options" on page 70.

Example

For example, to set both the native and Java stacks for the application server to a maximum of 512K, use the following command line:

```
SilverServer +Xss512k +Xoss512k
```

NOTE: Increasing any of the default stack size values will increase the amount of virtual memory allocated per thread. Virtual memory is a finite resource, albeit a large one (in a 32-bit operating system such as Windows NT, processes can address up to 2G of virtual memory). Increasing the per-thread virtual memory requirement will reduce the number of threads that can be created. It is important to realize that this could reduce the number of simultaneously connected users that the server is able to support (since the server uses one thread per connected client).

Miscellaneous issues

This section describes some uncategorized issues you may need to address.

Server appears to be hung

If the application server seems to be hung or in a loop, you can generate a listing for each thread with a stack trace. Doing this does not stop the server.

In NetWare Make sure the Logger buffer size is set large enough to handle the thread stack information. (You'll want to set the Recall and Layout to large numbers.)

Determine the JVM ID of the application server process:

```
java -show
```

To generate the stack trace:

```
java -showstacksXXXX
```

where *XXXX* is the JVM ID of the application server process.

To see the stack trace, you can:

- Use the Logger screen
- Use the GUI version of the Console Log (where you can save to a file)
- Go to http://hostname:8008 on a browser and use the Remote Manager to view the logs

In UNIX Determine the process that the application server is running under:

```
ps -all | grep Silver
```

Issue the following command:

```
kill -3 SilverServer_process_ID
```

The application server lists each thread with a stack trace in the window where the server was started from.

In Windows In the window where you started the server, press **Ctrl+Break**. The application server lists each thread with a stack trace.

Socket exceptions

You may receive a Socket Exception message in your NT application log. Typically, this is not a problem: it usually indicates that a client has unilaterally closed a socket. Browsers such as Internet Explorer frequently do this when the connection has been idle for a while, and it will show up as a Socket Exception in the server's console when running with debugging.

You can usually ignore such warnings; they simply reflect a normal situation.

III Appendixes

This part describes some miscellaneous topics concerning administering the Novell exteNd Application Server

A The httpd.props File

This appendix describes the httpd.props file and has these sections:

- About the httpd.props file
- Server properties

About the httpd.props file

The Novell exteNd Application Server maintains most of its properties in the internal AgProperties system table (for more information, see Appendix C, "System Tables and URLs"). You set these properties using the SMC or the Administration API (you don't access AgProperties directly).

However, some properties are needed at server startup time and are therefore stored externally so they are available when the server starts. These properties are stored in the **httpd.props** configuration file, which is located in the server's **\Resources** directory. You can change these settings either by modifying the httpd.props file or by using the SMC, which updates the httpd.props file when the change is saved.

NOTE: Whenever possible, it is best to make changes in the SMC rather than directly in the httpd.props file.

IMPORTANT: **Always stop the server before editing the file**. You can use the **Stop** button in the SMC. For more information, see "Shutting down the application server" on page 74.

Server properties

The following properties appear in the default httpd.props file and are listed in alphabetical order. (There are additional httpd.props properties that are optional and rarely used. They do not appear in the following table, but are described in the appropriate sections of the documentation.) All properties are case-sensitive.

NOTE: All property names begin with **http-server**.

Property/panel in SMC where settable	Description/default
com.sssw.db.dbplatforms NOTE: Not settable in SMC or API	Database platforms configuration file location **Default:** \\Resources\\platforms.dbl in the server's installation directory NOTE: **Do not change this value.**
com.sssw.orb.orbkey SMC: Configuration/General	ORB to use **Default:** ObjectEra_Jbroker

Property/panel in SMC where settable	Description/default
com.sssw.orb.orbplatforms NOTE: Not settable in SMC or API	ORB platforms configuration file **Default:** \\Resources\\orbs.dbl in the server's installation directory NOTE: **Do not change this value.**
com.sssw.srv.agent.debug SMC: Configuration/Advanced/Debug	Prints information about the execution of each running Web application. The number assigned (1-5) represents the level of verbosity for the messages. Enter **0** to disable. 📖 See "Low-level debugging" on page 219. **Default:** 0
com.sssw.srv.filecache.dir NOTE: Not settable in the SMC	Specifies the location of the filecache directory. The filecache directory contains the JAR files that the J2EE classloader is serving plus the contents of all WAR files. The default value is the \filecache directory in the application server's install directory.
com.sssw.srv.loader.debug SMC: Configuration/Advanced/Debug	Prints information about the execution of each ClassLoader. The number assigned (1-5) represents the level of verbosity for the messages. Enter **0** to disable. 📖 See "Low-level debugging" on page 219. **Default:** 0
com.sssw.srv.client.debug SMC: Configuration/Advanced/Debug	Prints debug messages for client connections to the server console. The number assigned (1-5) represents the level of verbosity for the messages. Enter **0** to disable. 📖 See "Low-level debugging" on page 219. **Default:** 0
com.sssw.srv.ContentCache.Disk.Directory NOTE: Not settable in SMC or API	Location of content cache **Default:** \\temp\\ContentCache in the server's installation directory
com.sssw.srv.http.authHeaderRealm SMC:Security/General	A String that will be displayed as the security realm in a server login dialog. This value is also passed on the WWW-Authenticate response header sent to the client. Requires a server reboot. **Default:** Novell exteNd Application Server
com.sssw.srv.http.listen_admin com.sssw.srv.http.listen_rt SMC: Configuration/General	Whether the server listens on any or all of the HTTP **administration** or **runtime** ports. Default is true, meaning that the server listens on the HTTP port. False means that the server will not listen on the HTTP port. 📖 See "Setting up separate ports" on page 79. **Default:** true
com.sssw.srv.http.webmaster SMC: Configuration/General	SilverMaster data source name **Default:** Set at installation
com.sssw.srv.https.listen_dsa_admin com.sssw.srv.https.listen_dsa_rt SMC: Security/Server Security	Whether the server listens on any or all of the HTTPS DSA **administration** or **runtime** ports 📖 See "Setting up separate ports" on page 79. **Default:** false

Property/panel in SMC where settable	Description/default
com.sssw.srv.https.listen_rsa_admin com.sssw.srv.https.listen_rsa_rt SMC: Security/Server Security	Whether the server listens on any or all of the **administration** or **runtime** HTTPS RSA ports 📖 See “Setting up separate ports” on page 79. **Default:** false
com.sssw.srv.https.port_dsa_admin com.sssw.srv.https.port_dsa_rt SMC: Security/Server Security	The HTTPS DSA **administration** and/or **runtime** port. At initialization time, the server will bind its accepting socket to the host it runs on and to the provided port. The server will use a DSA/Diffie-Hellman certificate and encryption algorithms for SSL on this port. 📖 See “Setting up separate ports” on page 79. **Default:** 444
com.sssw.srv.https.port_rsa_admin com.sssw.srv.https.port_rsa_rt SMC: Security/Server Security	The HTTPS RSA **administration** and/or **runtime** port. At initialization time, the server will bind its accepting socket to the host it runs on, and to the provided port. The server will use an RSA certificate and encryption algorithms for SSL on this port. 📖 See “Setting up separate ports” on page 79. **Default:** 443
com.sssw.srv.jms.debug NOTE: Not settable in SMC or API	Prints Novell exteNd JMS server-related debug messages to the server console. For basic debugging, specify **1**. For deeper debugging, specify a number greater than 1. Specify **0** to disable. 📖 See “Running the JMS server” on page 87. **Default:** 0
com.sssw.srv.jmsServerLaunch NOTE: Not settable in SMC or API	Whether the application server tries to start the JMS server when starting up. To automatically start the JMS server, specify **true**. Otherwise, specify **false**. When you install the application server, the installation program asks if you want to configure the JMS server and then sets this property according to your response. 📖 See “Running the JMS server” on page 87. **Default:** false (if jmsServerLaunch property is removed)
com.sssw.srv.loader.debug NOTE: Not settable in SMC	Whether to turn on ClassLoader-related debugging messages that display in the server console. Values range from 1 to 5; 5 gives the most detail. Enter 0 to disable. **Default:** 0
com.sssw.srv.logger SMC: Configuration/ General	The logging class **Default:** com.sssw.srv.http.AgLogger
com.sssw.srv.logger.logging SMC: Configuration/ General	Whether to log every standard HTTP client request to the server **Default:** false
com.sssw.srv.logger.errlogging SMC: Configuration/General	Whether to turn on error logging **Default:** true
com.sssw.srv.logger.errorlogname SMC: Configuration/General	The name of the error log file (if you are logging to a file) **Default:** errlog

Property/panel in SMC where settable	Description/default
com.sssw.srv.logger.logname SMC: Configuration/General	The name of the HTTP log file (if you are logging to a file) **Default:** log
com.sssw.srv.logger.tracelogging SMC: Configuration/General	Whether to turn on trace logging **Default:** false
com.sssw.srv.logger.tracelogname SMC: Configuration/General	The name of the trace file (if you are logging to a file) **Default:** traces
com.sssw.srv.logger SMC: Configuration/General	Java class to do the logging **Default:** com.sssw.srv.http.AgLogger
com.sssw.srv.nameServicePort SMC: Configuration/General	The port of the name service the server is using **Default:** 54890
com.sssw.srv.port_admin com.sssw.srv.port_rt SMC: Configuration/General	Server HTTP **administration** and/or **runtime** port 📖 See "Setting up separate ports" on page 79. **Default:** 80
com.sssw.srv.server NOTE: Not settable in SMC or API	Application server protocol version **Default:** exteNd Application Server/*n.n*
com.sssw.srv.sminit NOTE: Not settable in SMC or API	Location of internal sminit.props file, used by SilverMasterInit (do not edit this file) **Default:** the server's \\Resources\\sminit.props
com.sssw.srv.sql.debug SMC: Configuration/Advanced/Debug	Prints debug messages for SQL database calls to the server console. The number assigned (1-5) represents the level of verbosity for the messages. Enter 0 to disable. **Default:** 0
com.sssw.srv.SupportNTLocalGroup s NOTE: Not settable in SMC (settable as PROP_SUPPORT_NT_ LOCAL_GROUPS in AgiAdmServer and AgiAdmCluster)	Whether the application server uses NT local groups for authentication **Default:** true 📖 See "Speeding NT authentication" on page 127.
com.sssw.srv.system.out.log.allowed	Whether the application server when run as an NT service will log output to System.out and System.err **Default:** False
com.sssw.srv.system.out.log.file	The file to which the application server when run as an NT service will log output to System.out and System.err (if the property system.out.log.allowed is True) **Default:** the server's \\temp\\SilverServerSysOut.txt
Jdbc.LDSKey SMC: Configuration/Databases	JDBC driver for SilverMaster **Default:** Set at installation
Jdbc.URL SMC: Configuration/Databases	The database URL. The driver uses it to connect to the SilverMaster database. The URL is driver-specific. See your driver documentation for more information. **Default:** Set at installation

Property/panel in SMC where settable	Description/default
Jdbc.URL.Attributes SMC: Configuration/Databases	Any extra attributes to set for the JDBC driver. The syntax is driver-specific. See your driver documentation for more information. **Default:** Set at installation

B SNMP Agent

This appendix describes how you can set up and test SNMP to monitor the Novell exteNd Application Server. It has these sections:

- About SNMP
- SNMP implementation overview
- Setting up SNMP for the application server
- Setting up access from your SNMP Management node

About SNMP

SNMP (Simple Network Management Protocol) is a protocol used to remotely manage and control nodes on a TCP/IP network. Using SNMP, one workstation running management software can monitor information being collected by routers, servers, and other workstations on the system. This information is used to determine the performance integrity of the network.

NOTE: The SNMP implementation currently runs on the Windows NT and Windows 2000 platforms only. It does not allow the SNMP service to control the application server.

SNMP implementation overview

The application server implements SNMP using the following components:

Component	Description
snmp_options.props	File that defines the following settings used by the AgSNMPGetStats servlet: ◆ **StatisticsUpdateInterval**—Number of seconds to wait before updating the statistics file. Default is 120. ◆ **WriteStatisticsEnabled**—Whether to write server statistics and the update interval to AgSNMP.props; 0 for false, 1 for true. Default is 1. ◆ **StatisticsDebug**—Whether to send debugging messages to the server console; 0 for false, 1 for true. Default is 0. The file is located in server's \Resources directory.
AgSNMPGetStats servlet	A load_on_startup servlet that must be deployed to the SilverMaster database. It is responsible for updating application server statistics in the AgSNMP.props file.
AgSNMP.props	File that the AgSNMPGetStats servlet writes the server statistics to at the interval specified in snmp_options.props. (WriteStatisticsEnabled must be set to 1 in snmp_options.props for the statistics to be written.) The file is located in server's \Resources directory.

Component	Description
SNMP extension agent (AgSNMP50.dll)	Implements the Windows NT SNMP Application Program Interface (API). The SNMP extension agent reads the application server statistical information from AgSNMP.props.

For a list of statistics and object identifiers, see "Setting up access from your SNMP Management node" on page 247.

How the components work

An application server load_on_startup servlet (AgSNMPGetStats) updates the AgSNMP.props file at a scheduled interval. When the servlet is loaded, the init() method fires. This method:

1. Gets registry information from the application server to determine the server's installation path.
2. Reads the snmp_options.props file in the server's **Resources** directory. This file contains settings such as where to print debugging messages as well as the statistics update interval.
3. Starts a timer task (that runs every minute) to check whether the file update interval in the snmp_options.props file has changed.
4. Starts a timer task that runs on the specified file update interval to build the statistics data and write it to the AgSNMP.props file in the server's **Resources** directory.

On an SNMP GET request, if the update interval has elapsed, the extension agent updates the MIB data by accessing the registry to get the server's path and reading the AgSNMP.props file in the server's **Resources** directory. Otherwise, it returns the value that was stored the last time the file was read. If the timestamp in the file is not updated within the given interval, the Server Responding status is set to false—indicating a possible problem with the application server.

NOTE: The extension agent does not use a timer to determine whether the update interval has changed. If the interval is decreased by a substantial amount, it can give a false Server Responding status. When the interval value is decreased, you should stop and restart the SNMP service.

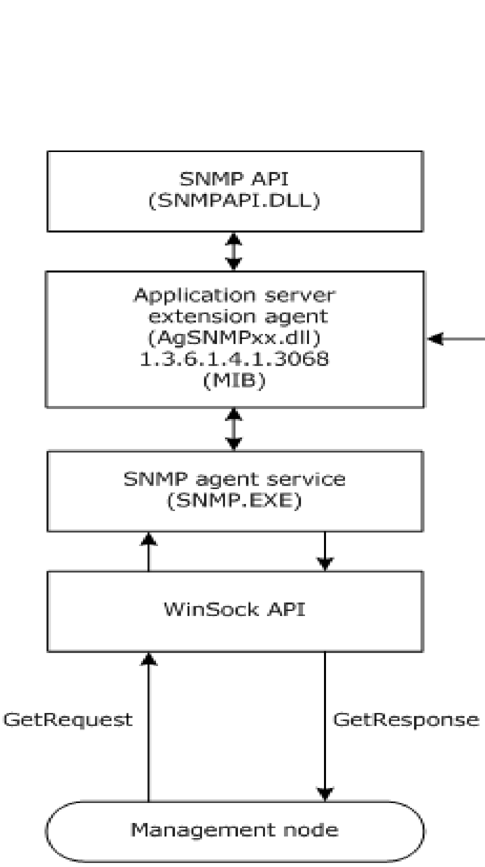
Process flow and terminology

The following terms (shown in the following figure) are used in an SNMP-enabled architecture:

Term	Description
Management node	The workstation or server running one or more network management processes. These processes are usually software applications that gather information from the managed nodes, or SNMP agents. Examples of management node software include Unicenter TNG from Computer Associates, OpenView from Hewlett-Packard, and Tivoli from IBM.
Managed Information Bases (MIBs)	The hierarchical map of all managed objects and how they are accessed.
Managed objects (MIB objects or variables)	The collection of objects that describe the SNMP managed node to the management node. This data is defined with a specific set of attributes that are manipulated using the standard SNMP operations Get, GetNext, and Set.
Object identifier (OID)	A unique identifier for a MIB variable. An OID is the location of a managed object within a MIB namespace. The OID of a MIB object is also referred to as the object's identity or registration.

Term	Description
SNMP agent	Software or firmware that runs as one or more processes on a managed server. The SNMP agent provides management services by collecting and returning management information requested by the management node. An SNMP agent can be read-only, or it may allow the management node to control or alter the node it is managing. An SNMP agent may also generate traps, which are unrequested notifications of events.
Extension agent (Subagent)	A DLL that implements a set of registered managed objects defined in a MIB module and communicates with the SNMP service using the SNMP API.

The following figure shows how the components work within the SNMP framework:



Setting up SNMP for the application server

Setting up SNMP as a service involves the following basic steps:

1 Installing the SNMP software as a service

2 Installing the application server

3 Deploying the AgSNMPGetStats servlet

4 Testing the SNMP program

Installing SNMP as a service

If you are installing the SNMP software service on a machine that is currently running the application server, you need to first stop the application server.

📖 For more information about installing SNMP as a service, see your operating system documentation.

Installing the application server

If you haven’t yet installed the application server, use the installation program to install it on the machine to be managed. If you choose to reinstall the application server, you must first stop the SNMP service; otherwise, the install program will not be able to write over the AgSNMP50.dll file.

The server’s installation program takes care of the required registry key entries and places the agent in the correct location. The required registry entries are as follows:

Item	What to specify
Key	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SNMP\Parameters\ExtensionAgents
Name	Ag*version*SNMP
Value	Novell\eXtend\AppServer\version\SNMP\ExtensionAgents\AgSNMPAgent\CurrentVersion
Key	HKEY_LOCAL_MACHINE\SOFTWARE\Novell\eXtend\AppServer\version\SNMP\ExtensionAgents\AgSNMPAgent\CurrentVersion
Name	Pathname
Value	*server*'s \bin\AgSNMP*version*.dll

Deploying the AgSNMPGetStats servlet

A number of files are provided to build and deploy the AgSNMPGetStats servlet. They are located in the server’s servertools\snmp directory:

File	File name(s)
A readily deployable EAR file	SilverGetStats.ear
A batch file to deploy the EAR to the SilverMaster	deploySilverGetStats.bat
A batch file to undeploy the EAR from the SilverMaster	deleteSilverGetStats.bat
A deployment plan	SilverGetStats_depl_plan.xml)
Project files to build the WAR and EAR files	SilverGetStatsWar.spf SilverGetStatsEar.spf
Source files to build the WAR and EAR files	—

To deploy the EAR, run the deploySilverGetStats.bat file, passing the server name and the SilverMaster name.

You can also rebuild the EAR and deploy it using the project files provided.

Testing the program

The application server provides a tool that allows you to test the SNMP installation from a DOS prompt.

➢ **To test the SNMP extension agent:**

1. Stop and restart the **SNMP Service**.
2. Open a DOS window.
3. Change to the server's **\bin** directory.
4. Run the batch file **SilverSNMPGetStats.bat**, passing the server name and SNMP parameter name. To get a list of available parameters, type:

```
SilverSNMPGetStats -?
```

Setting up access from your SNMP Management node

The Object Identifier is composed of an enterprise ID and the OID identifying each MIB. You will need the following information to set up access to the application server's OIDs from your SNMP Management node:

Item	Description
Private Enterprise ID	A unique number assigned to a company. The server's private enterprise ID is 3068.
Object Identifier (OID)	The server's host name OID is 1.3.6.1.4.1.3068.1.7.7.1.0.

The following are the application server's OIDs. When accessing MIB data using **SNMPTool.exe**, you must precede the OID with a period:

Statistic description	OID	Data type
Host Name OID	1.3.6.1.4.1.3068.1.7.7.1.0	OCTET STRING
Server Revision OID	1.3.6.1.4.1.3068.1.7.7.2.0	
Server Start Time OID	1.3.6.1.4.1.3068.1.7.7.3.0	
Data Time Snapshot OID	1.3.6.1.4.1.3068.1.7.7.4.0	
Maximum Requested URL OID	1.3.6.1.4.1.3068.1.7.7.5.0	
Minimum Requested URL OID	1.3.6.1.4.1.3068.1.7.7.6.0	
Server Load OID	1.3.6.1.4.1.3068.1.7.7.7.0	INTEGER
Free Thread Count OID	1.3.6.1.4.1.3068.1.7.7.8.0	
Idle Thread Count OID	1.3.6.1.4.1.3068.1.7.7.9.0	
Total Thread Count OID	1.3.6.1.4.1.3068.1.7.7.10.0	
Hit Count OID	1.3.6.1.4.1.3068.1.7.7.11.0	
Mean Request Time OID	1.3.6.1.4.1.3068.1.7.7.12.0	
Max Request Time OID	1.3.6.1.4.1.3068.1.7.7.13.0	
Min Request Time OID	1.3.6.1.4.1.3068.1.7.7.14.0	
Emitted Bytes OID	1.3.6.1.4.1.3068.1.7.7.15.0	

Statistic description	OID	Data type
Free Memory OID	1.3.6.1.4.1.3068.1.7.7.16.0	Counter
Total Memory OID	1.3.6.1.4.1.3068.1.7.7.17.0	
Garbage Collection Count OID	1.3.6.1.4.1.3068.1.7.7.18.0	
Idle Sessions OID	1.3.6.1.4.1.3068.1.7.7.19.0	INTEGER
Total Sessions OID	1.3.6.1.4.1.3068.1.7.7.20.0	
Server Responding OID	1.3.6.1.4.1.3068.1.7.7.21.0	OCTET STRING

C System Tables and URLs

The Novell exteNd Application Server stores system data in a database called the SilverMaster (the SilverMaster is described in Chapter 4, “Data Source Configuration”). This appendix provides a listing of system tables and database URLs:

- Application server internal system tables
- Database URLs

NOTE: The items listed are reserved for the application server’s use. This listing is provided for informational purposes only.

Application server internal system tables

The following are the application server’s internal tables. The information in these tables controls major server functions such as security and server clustering for load balancing. Some tables are used in each database connected to the server. The remainder of the tables exist in the SilverMaster only.

Tables in all databases

The following tables exist in or in conjunction with each database connected to the application server (including SilverMaster).

Database system table	Purpose
AgAccessRights	Contains security access information on resources.
AgAgents	Contains information on EJB JARs and WARs (for server-start servlets) deployed to the server.
AgContents	Contains the contents of application resources including all of the deployed EARs, WARS, and JARS.
AgInfo	Contains a catalog of known databases and their system information.
AgResources	Manages details of all objects stored in application server databases.

Tables only in SilverMaster

The following tables exist in the SilverMaster database only.

SilverMaster table	Description
AgCacheMgr	Contains Cache Manager information.
AgCacheMgrGroup	Contains Cache Manager group information.
AgCertificates	Contains certificate information.

SilverMaster table	Description
AgCluster	Contains cluster information. Used with load balancing software.
AgClusterEnv	Contains cluster information. Used with load balancing software.
AgDispatcherMgr	Contains Dispatcher information. Used with clusters.
AgDispatcherMgrGrp	Contains Dispatcher group information. Used with clusters.
AgErrorLog	Contains error log information.
AgGroupMembers	Contains a list of members of groups in UUID format.
AgLoadMgr	Contains Load Manager information. Used with clusters.
AgLoadMgrGroup	Contains Load Manager group information. Used with clusters.
AgLog	Contains application server log information.
AgObjects	Contains object information.
AgPasswords	Contains application server password information.
AgProperties	Contains application server properties.
AgPrincipals	Contains a list of application server users and groups in UUID format.
AgServer	Contains application server information.
AgSessBeans	Used for J2EE failover support and stateful session bean passivation. Contains HTTP session state and stateful session beans (local and remote).
AgSessSubjects	Used for J2EE failover support. Contains login state (via encrypted Subject info) for HTTP session state if session is logged in.
AgTraceLog	Contains trace log information.
AgUserLicense	Not used.

Database URLs

The following table lists the directory structure for items located in /***database***/**SilverStream/**. This listing is provided for informational purposes only.

Database URL	Description
Administration	Server administration settings
Classes	(Not used)
ClusterAdmin	Administrative functions (server clusters only)
Downloads	Temporary communication endpoints
ErrorLogs	Error messages
Login	User login access
Logout	Session logout

Database URL	Description
Meta/... ◆ Agents ◆ Certificates ◆ Entities ◆ Forms ◆ Licenses ◆ Reports ◆ ServerCertificate ◆ Tables ◆ UUIDTranslator ◆ Views ◆ Webbases	Corresponds to... ◆ (Not used) ◆ Certificates resource ◆ Table names resource ◆ (Not used) ◆ (Not used) ◆ (Not used) ◆ Certificates resource ◆ Table names and contents ◆ (Internal resource) ◆ (Not used) ◆ Databases
Objectstore/	Uploaded media files
Pages	HTML pages (active presentation & static)
Resources	Directory of system files
Security	Endpoint for security administration
Sessions	Displays current session information, including session number, user, host, and state
SilverJ2EEClientInstall	SilverJ2EEClient installation
Statistics	Displays server statistics
Timestamps	Displays timestamped events (internal only)
VersionCheck	Displays the application server’s version number

D WSI Directives Reference

This appendix includes two sections:

- Apache WSI directives reference
- AgWSI.conf file reference (for IIS and iPlanet configurations)

Apache WSI directives reference

This section describes each configuration setting, specifies which settings are required, and provides defaults and examples. Settings can apply at these levels:

Settings at these levels	Apply to	Where to specify in the httpd.conf file
Server-wide	All application servers defined in the httpd.conf file	Global section
Single server	A single application server	Any section
Directory	A single directory for a single application server	LocationMatch directive

SetHandler

Description

Required. Per directory setting. Must be placed in a LocationMatch directive container.

Notifies Apache that the WSI module will handle any URL that matches the LocationMatch directive.

Example

```
SetHandler wsi-handler
```

WSICleanupInterval

Description

Optional. Per connection pool setting. Must be placed in the WSIConnPool container.

Defines the time interval (in seconds) that the WSI module will look for idle connections in the connection pool. Connections that have been idle longer than the idle timeout value are disconnected.

Default

300 seconds (5 minutes)

WSIConnPool

Description.

Optional. Per directory setting.

A container directive whose contents define a named connection pool. The default named connection pool name is cp01. You can define multiple connection pools in the same httpd.conf file.

Example

```
<WSIConnPool cp01>
.
.
.
</WSIConnPool>
```

WSIHost

Description

Required for creating a connection pool. Must be in a LocationMatch container if not using a connection pool.

Specifies the hostname or IP address on which the application server is running.

This directive is ignored if a connection pool is already defined (using the WSIConnPool directive) for the same LocationMatch directive container.

Default

localhost

Example

```
WSIHost      alaska.novell.com
```

WSIIdleTimeout

Description

Optional. Per connection pool setting.

Specifies the maximum idle time (in seconds) for SSL and non-SSL connections to the application server.

Default

600 seconds (10 minutes)

Example

```
WSIIdleTimout    450
```

WSIMaxConns

Description

Optional. Per connection pool setting.

Specifies the maximum number of non-SSL connections for the connection pool.

Default

20

Example

```
WSIMaxConns   35
```

WSIMaxSslConns

Description

Optional. Per connection pool setting.

The maximum number of SSL connections for the connection pool.

Default 20

Example `WSIMaxSslConns   35`

WSIPort

Description **Required if defining a connection pool**. Per directory if connection pooling is not specified.

Specifies the application server’s HTTP port. This directive is ignored if a connection pool is already defined (using the WSIConnPool directive) for the same LocationMatch directive container.

Default 80

Example `WSIPort   10800`

WSISslPort

Description **Required if defining a connection pool**. Per directory if connection pooling is not specified.

Specifies the application server’s HTTPS port. This directive is ignored if a connection pool is already defined (using the WSIConnPool directive) for the same LocationMatch directive container.

Default 443

Example `WSISslPort   10443`

WSITraceLevel

Description Optional. Server-wide setting.

Specifies the level of tracing for the WSI module. Higher values produce more detailed trace information. The WSI module can generate a lot of trace information when set at the higher debug levels. In general, do not set the trace level higher than 3 unless there is a specific reason for doing so. Valid settings are:

Level	Trace event	Description
0	CRIT	Critical failures
1	ERROR	Errors
2	WARNING	Warnings
3	INFO	Informational messages
4	DEBUG	First--level debug
5	DEEP	Second--level debug
6	DEEPER	Third--level debug
7	DEEPEST	Fourth--level debug

Default 3

Data type Integer

Example `WSITraceLevel    6`

WSITraceMode

Description Optional. Server-wide setting.

Determines how the WSI module writes trace information to the trace output files. The valid values are:

Value	Description
per-process	Each Apache process has its own WSI trace output file created in the WSITraceOutputDirectory with the name agapache.p*xxxx*.out (where *xxxxx* is the process ID).
per-request	A new WSI trace output file is created for each request processed by the WSI module. The output files are created in the WSITraceOutputDirectory with the name "agapache.p*xxxxx*.r*yyyyy*.out" (where *xxxxx* is the process ID and *yyyyy* is the request number). IMPORTANT: This setting was created for internal debugging purposes only, and should **not** be used in a production system.

Default per process

Example

```
WSITraceMode    per-process
```

WSITraceModuleWidth

Description Optional. Server-wide setting.

Specifies the maximum width of modules or function names in WSI trace events.

Default 16

Example

```
WSITraceModuleWidth    20
```

WSITraceOutputDirectory

Description Optional. Server-wide setting.

Specifies the file system directory where the WSI module creates trace files.

If specified, it must correspond to an existing directory with write permissions granted to the Apache process.

Default

Operating system	Default
NetWare	sys:/tmp
UNIX	/tmp
Windows	c:\temp

WSITracePadModules

Description. Optional. Server-wide setting.

Specifies the formatting of the trace event names listed in the WSI trace.

Values are on or off. When set to on, module or function names shown in the WSI trace events are space-padded. This formatting makes trace files easier to read (but larger), because the trace records will line up on the same column.

Default on

WSITraceTimestamps

Description Optional. Server-wide setting.

Values are on or off. When set to on, WSI trace events include timestamps.

Default on

WSIWatcherInterval

Description Optional. Server-wide setting.

Specifies the time interval (in seconds) between iterations of the connection pool watcher facility. The watcher prints information about the connection pool to the trace files located in the WSITraceOutputDirectory.

When 0 is specified, the watcher thread is disabled.

Default 0

WSIUrl

Description Optional. Per directory setting.

Specifies a relative URL that is substituted for the URL fragment specified in the LocationMatch directive.

Example For example, given the following directives:

```
<LocationMatch /foo>
SetHandler wsi-handler
WSIUrl /bar
</LocationMatch>
```

A request for an URL of the form http://foo/whatever_follows will be handled by the WSI module as http://bar/whatever_follows.

If this directive is not specified, the URL substitution is not performed.

AgWSI.conf file reference

This section describes each configuration setting, specifying which settings are required and providing defaults and examples.

Connection.http.max

Description

Optional.

Connection.http.max is the maximum number of concurrent **nonsecure** HTTP connections between the WSI and the application server.

Usage

If you have created and specified a **WSI.error.url** file, users will be notified if the connection pooling limit is exceeded. The WSI reuses socket connections between itself and the application server.

Default

```
Connection.http.max=100
```

Connection.https.max

Description

Optional.

Connection.https.max is the maximum number of concurrent **secure** HTTPS connections between the WSI and the application server.

Usage

If you have created and specified a **WSI.error.url** file, users will be notified if the connection pooling limit is exceeded. The WSI reuses socket connections between itself and the application server.

Default

```
Connection.https.max=100
```

Connection.idle.time

Description

Optional.

Connection.idle.time specifies how often (in minutes) the WSI scans the connection pools for idle connections.

Default

```
Connection.idle.time=25
```

SECTION

Description

Optional.

SECTION names a configuration section. Each section contains the statements that specify the handling of a set of Web requests by an application server. If you want the Web server to direct different requests to different application servers, use multiple sections in the configuration file.

Usage

Each section must define all required settings. If an optional setting is not defined in a section, it will be given the default value (not the value defined in any other section).

Format

```
SECTION=label
```

For the label, enter something informative.

Examples

```
SECTION=abc_com
SECTION=xyz_com
```

For more info

For more information about the use of configuration sections, see "Directing requests to multiple application servers" on page 113.

SilverServer.host

Description

Required.

SilverServer.host is the name of the destination application server that services URL requests from the Web server.

Example

```
SilverServer.host=mysssw.myco.com
```

SilverServer.http.port

Description

Required if the application server HTTP port is not using the default port number for your operating system.

SilverServer.http.port specifies the **nonsecure** port of the destination application server. Use the value zero (0) to specify that the WSI should not forward any requests coming in on the nonsecure port.

Default

```
SilverServer.http.port=80
```

SilverServer.https.port

Description

Required if the application server HTTPS port is not using the default port number (443).

SilverServer.https.port specifies the **secure** port of the destination application server. Use the value zero (0) to specify that the WSI should not forward any requests coming in on the secure port.

Default

```
SilverServer.https.port=443
```

SilverServer.urls

Description

Required.

SilverServer.urls specifies which URLs will be forwarded to the application server. You must specify a new setting for each URL root you want forwarded to the application server.

Formats

There are two formats: Simple URL forwarding and URL forwarding with translation (masking).

Simple URL forwarding

Syntax:

```
SilverServer.urls=<Root_URL_to_Forward>
```

Examples:

```
SilverServer.urls=/myco
SilverServer.urls=/myDb
```

In the preceding examples, all URLs that begin with either /myco or /myDb will be forwarded to the application server. This includes:

```
http://myWebServer/myco/Sessions
http://myWebServer/myco/Pages
http://myWebServer/myDb/myco/Pages/MyPage.html
```

To forward **all** URLs to the application server, specify the following:

```
SilverServer.urls=/
```

URL forwarding with translation (masking)

Syntax:

```
SilverServer.urls=<URL_root_at_Web_server>=<translated_URL_root>
```

Example:

```
SilverServer.urls=/Pages=/myDb/myco/Pages
```

The first URL will be forwarded to the application server after the second URL is substituted for it. In this example, all URLs from the Web server that begin with /Pages will be forwarded to the application server with /myDb/myco/Pages substituted for /Pages. The URL sent to the Web server as:

```
http://myWebServer/Pages/MyPage.html
```

is forwarded to the application server as:

```
http://mySilverServer/myDb/myco/Pages/MyPage.html
```

WSI.auth.echo

Description Optional.

When a request sent to the Web server contains an HTTP authorization header, the WSI will send an HTTP header (called x-agwsi-Authorization) to the application server that echoes the value of the header when WSI.auth.echo is set to true.

This setting allows the application to retrieve the user login information when the user login has been masked with the **WSI.auth.user** command. For example, when a third-party product is performing authentication and authorization services, the WSI.auth.echo setting allows the application to retrieve the name of the user who logged in to the application and initiated the request.

Format The HTTP header will appear in the following (name/value) format:

```
x-agwsi-Authorization: Basic Base64EncodedUserName/Password
```

Default `WSI.auth.echo=false`

Usage The application server uses the AgiHttpServletRequest API to retrieve the authorization header.

WSI.auth.NTLM.remove

Description Optional. Used with IIS only.

Set WSI.auth.NTLM.remove to true if NT authentication is enabled for your IIS directories. Setting the value to **true** removes the NTLM authentication headers so users' requests can be successfully forwarded to the application server. For more information, see "Using IIS NTLM authentication with the WSI module" on page 117.

Default `WSI.auth.NTLM.remove=false`

WSI.auth.user

Description

Optional.

When WSI.auth.user is specified, the WSI module intercepts the authentication headers that will be forwarded to the application server and replaces them with the credentials of a single known user. Then it adds HTTP authentication headers to every request it forwards to the application server. Any existing authentication headers on incoming requests to WSI are replaced by the authentication setting.

Usage

To protect the security of the authentication settings, the user name and password are not stored in clear text in the AgWSI.conf file. You must run the AgWSIUser utility to generate the **WSI.auth.user** statement needed in AgWSI.conf to represent a given user and password. The AgWSIUser utility encrypts the user name and password in a form the WSI can read.

The authentication setting can be used to:

- Remove NTLM authentication headers added by IIS
- Distinguish requests that have been forwarded by the WSI to the application server

WSI.debug

Description

Optional.

WSI.debug specifies the WSI logging level. The WSI logs to the AgWSI.log file, which is stored in your WSI module directory.

Usage

Choose from these levels:

Level	Information logged
0	None
1	Each request method, URL, and whether it was processed successfully Any errors in the connection between the WSI module and the application server Connection pool cleanup messages
2	Level 1 information plus: ◆ Full HTTP response and reply headers and content lengths
3	Level 2 information plus: ◆ URL mapping results

To log the URLs arriving at the destination server, use the SMC to configure your application server debug value to 1 or 2. For more information, see “Low-level debugging” on page 219.

Default

```
WSI.debug=0
```

WSI.error.url

Description.

Optional.

WSI.error.url specifies a customized error page users will see if a WSI connection error occurs. If you do not create and specify a WSI error URL, users will see a generic browser notification when the WSI module cannot connect to the application server.

Usage

It is a good idea to create an HTML file that tells users about the problem and advises them to retry the URL connection later.

Specify the WSI error file name and its location on your Web server. You need to put your error page file in your Web server's directory structure, since the WSI redirects the browser to this page on your Web server.

Default

```
WSI.error.url=\myerror.html
```

WSI.host

Description

Optional.

WSI.host specifies the HTTP host header to filter when matching URLs to be forwarded to an application server. If this statement is not specified, the request host header is ignored and only URL matching is used as a filter.

Usage

This setting is used for multihomed Web server configurations, where the Web server is hosting multiple separate host names and the WSI needs to forward URLs to different application servers based on the request host name. The WSI.host setting can specify either the hostname alone or the hostname and the port number. If the port number is not specified, the WSI will accept any port number on the request's host header provided the host names match.

Examples

```
WSI.host=www.abc.com
WSI.host=www.abc.com:8080
```

For more info

For a sample use of this statement, see "Directing requests to multiple application servers" on page 113.

WSI.root.dir

Description

Required for IIS only.

WSI.root.dir is the WSI virtual directory the WSI module runs from. Since the WSI for IIS is both a filter and an extension, you need to specify the URL for the WSI in the IIS Web root (/wwwroot) directory structure.

Usage

You need to install the WSI module for IIS in a directory that is visible from the IIS Web root directory. If you install the WSI in a physical directory below the Web root, the WSI is automatically visible from within IIS. But if you install the WSI module in a directory that is outside the IIS root directory, you must create a virtual directory using the MMC so the WSI appears in the IIS directory.

You must set WSI.root.dir relative to the Web server root directory.

Examples

If you are installing the WSI into C:\Inetpub\wwwroot\agisapi (a physical directory beneath the Web root), specify WSI.root.dir=/agisapi.

If you are installing the WSI into a virtual directory such as C:\WSI, use the MMC to create a virtual directory such as /agisapi that maps to C:\WSI and then specify WSI.root.dir=/agisapi.

Default

```
WSI.root.dir=/agisapi
```

Index